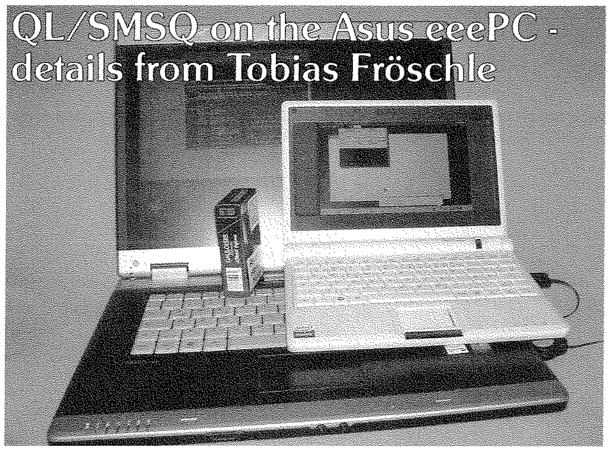
QL Today

Volume 12
Issue 4
June-August
2008

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

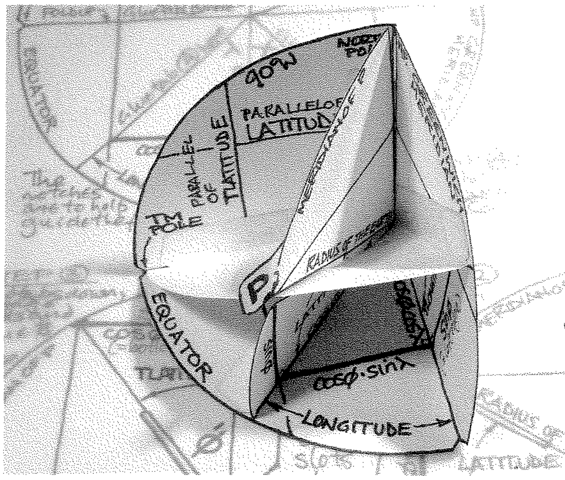


QL/SMSQ on the Asus eeePC - details from Tobias Fröschle

EARLY ISSUE!

Yes, we are early! The reason is on the reverse side of this magazine - there could be a QL show near you very soon - maybe even today or tomorrow!

www.QLToday.com



What's that? You'll find out by reading Hugh Rooms article and you can build it yourself!

TIME TO RENEW!

If you have not yet renewed with the renewal form found in the previous issue, then please do so now - same price as last volume - and as easy and with as many payment methods as never before. Renewal form in the shipping envelope of this issue! Thank you!!

The biggest Letter-Box ever!

QL Today Toolkits Cover Disk

New Cover Disk!

Dilwyn Jones has compiled another interesting cover disk for you...

Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software	Tel. +49 203 502011
Kaiser-Wilhelm-Str. 302	Fax +49 203 502012
47169 Duisburg	email: smsq@j-m-s.com
Germany	email: QLToday@j-m-s.com

English office:

Q Branch	Tel. +44 1273 430501
20 Locks Hill	Mobile +44 7836 745501
Portslade	Fax +44 1273 430501
BN41 2LB	email: qbranch@qbranch.demon.co.uk
United Kingdom	email: QLToday@j-m-s.com

Editor:

Geoff Wicks	Tel. +44 1332 271366
5b Wordsworth Avenue	email: gtwicks@btinternet.com
Sinfin	email: QLToday@j-m-s.com
Derby DE24 9HQ	
United Kingdom	

Co-Editor:

Bruce Nicholls	Tel +44 20 71930539
38 Derham Gardens	Fax +44 870 0568755
Upminster	email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA	email: QLToday@j-m-s.com
United Kingdom	

QL *Today* is published five times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Jonathan Hudsons web site where you find more information about lots of interesting QDOS software and INFOZIP at www.bigfoot.com/~jrhudson/

The deadline for the next issue is the 18th of August 2008

Editorial

by Geoff Wicks

Renewal time! This is the issue in which I report our statistics for the previous year, and this time they come with a bonus surprise for Quanta members.

This survey covers five issues totalling 294 A4 pages or almost 59 pages per issue. Of these 85% or 50 pages were editorial content, a higher proportion than the previous year. On average there were nearly 5 pages of news per issue. 15 different writers contributed to QL Today and a further 3 people contributed to the letter box.

I also make an analysis of the Quanta Magazine each year and the differences between volume 23 and volume 24 are striking. The average size of the magazine has increased from 34 to 41 pages and editorial content from 25 to 31.5 pages. The number of contributors has risen from 9 to 14. Well done Quanta committee!

Last time I wrote in QL Today "I am amazed that after nearly a quarter of a century QL-ers are still able to produce such a variety of original and interesting material". A good opportunity to thank all our writers without whose loyalty and enthusiasm QL Today would not exist.

QL Today is produced by a team and my thanks go to the other members of the team, Jochen Merz, Bruce Nicholls and Roy Wood. Our move to quarterly publication has proved to be worthwhile and it has been easier for us to stick to a regular publishing schedule. Events during the year have given me a greater insight into the work that all team members do.

Last year in this editorial I wrote:
"QL Today believes in healthy competition and we would like to see the Quanta committee set up a news service, both in print and on the web, to rival that of ours."

When I raised this issue at the 2007 Quanta AGM and offered QL Today's help in training a "news reporter" several members openly jeered. However Quanta's committee took the proposal seriously and have now appointed our ex-editor, Dilwyn Jones, as their first news editor. QL Today highly approves of this appointment.

Shortly after his appointment Dilwyn wrote:
"QL Today has always managed decent enough news coverage under both editors (and the few issues Jochen did alone), so let's see what we can achieve for Quanta now. While I foresee this friendly rivalry and good natured banter between the two organisations continuing, at the end of the day we should be able to co-operate well when it matters too. And I'm sure Geoff will keep us on our toes ;-)"

Two years ago I wrote in an editorial:
"A strong Quanta Magazine would be good not only for Quanta but also for the whole QL community.
And, believe it or not also for QL Today."

I am sure Dilwyn will be keeping us on our toes. That can only be good for QL-ers.

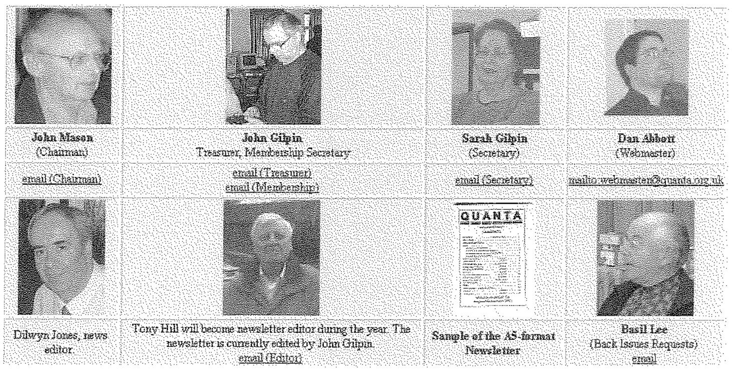
QUANTA'S Crunch Year

2008 could be Quanta's make or break year. Nowhere was this more apparent than at the recent Manchester AGM. On the one hand there was a much better attendance than in recent years, positive reports from the officers and two new committee members. On the other hand the meeting approved a detailed timetable to wind up Quanta next year should it prove impossible to find a replacement for current chairman John Mason.

Towards the end of 2007 both QL Today and the Quanta Magazine reported that unless new members were elected to the committee Quanta's closure could not be far off. Following these warnings there were informal discussions at the Solihull workshop during which the names of two possible new committee members emerged. John Mason told QL Today he sensed a revived interest in Quanta.

By the end of the year the optimism appeared misplaced. John Mason reported in the Quanta Magazine that no one had responded to his appeal for new committee members. However in February QL Today heard of members who had been nominated for the committee, but numerous off the record contacts with members of the existing committee failed to give us a clear picture of what was happening. QL Today could only report that Quanta would have some good news for its members. We now understand the situation was confusing with one clear nomination, a second member withdrawing his nomination, a third late nomination and a fourth member expressing an interest in taking a more active role in Quanta, but outside the committee.

Eventually all the members of the previous committee were prepared to remain in their present posts and they welcomed two new colleagues. Dilwyn Jones was elected at the AGM and Tony Hill was co-opted at a committee meeting held immediately afterwards. Dilwyn will be taking on the new responsibility of news reporter and Tony the editorship of the Quanta Magazine. Our illustration, taken from Dilwyn's website, shows, among other things, the entire committee.



John Mason (Chairman)	John Gilpin Treasurer, Membership Secretary	Sarah Gilpin (Secretary)	Dan Abbott (Webmaster)
email (Chairman)	email (Treasurer) email (Membership)	email (Secretary)	mailto:webmaster@quanta.org.uk
Dilwyn Jones, news editor.	Tony Hill will become newsletter editor during the year. The newsletter is currently edited by John Gilpin. email (Editor)	Sample of the A5-format Newsletter	Basil Lee (Back Issues Requests) email

An enlarged committee means Quanta can make progress on several projects that have been in abeyance. QL Today knows of three areas where Quanta wants to make changes. These are the website; archiving of the magazine; and reorganisation of the library.

Quanta launched a new website last year and wishes to further develop the site. Among various possibilities is the creation of a "members only" area containing more articles from the magazine than are available on the public area; some programs from the library; and even some form of forum. Quanta's webmaster, Dan Abbott, would also like to see more participation by the subgroups. Unfortunately the website has had problems in the last year including out of date content. A job change, temporarily involving frequent lengthy journeys, has restricted the time available to the webmaster for Quanta duties. At the moment it is uncertain how far the website will make use of the newly appointed news reporter. There are still questions to be answered about who is responsible for the site's editorial content and what content can be made available to non-members as well as members.

A couple of months ago John Gilpin appealed on the QL-users email group for back editions of the Quanta Magazine that Quanta no longer possesses. QL Today understands Quanta has embarked on a structural archiving of the magazine that will be available to all members. At the moment we have no further details.

The library has long been a source of concern for Quanta. Much of the software in the library is outdated and even obsolete. Quanta was hoping to find a committee member specifically to reorganise the library but this has proved impossible. However, QL Today understands there may be a member who is prepared to undertake some library work, but this will not be as a committee member. At the moment we cannot say whether work on the library can go ahead.

In spite of the progress Quanta has made, the organisation still remains in danger. The most immediate crisis is chairman, John

Mason, who under the constitution has to stand down next year. As yet there is no obvious successor. For some time members of the committee have been canvassing several people, but have been unsuccessful in finding a replacement. Similar problems will occur in 2010 and 2011 when John Gilpin and Sarah Gilpin respectively have to step down.

At the recent AGM members approved a procedure for winding up the organisation. Should there be no candidate for chairman at next year's AGM, then several sections of the constitution will be suspended to allow the committee three months to find a successor. If there is still no chairman then a special general meeting to wind up Quanta has to be called 3 months later.

The crunch date for Quanta could be October 2009.

TFS Clarification

In the last issue our lead news story was the decision of **Tony Firshman** to become less active in the QL community. This was because of pressures on his time elsewhere. Although Tony accepts our story was factually accurate, he has objected to the headline that implied he was leaving the QL scene. Some readers appear to have read the headline but not the article.

QL Today accepts our headline was misleading.

In a statement on the QL-users email group, editor Geoff Wicks wrote:

"The QL Today article does give an accurate description of Tony's situation, but in one way we have contributed to the misunderstanding.

The headline "TFS bows out" does give a misleading impression and Tony's suggestion to me of "TFS takes a back seat" or similar would have been better. I take full responsibility for this misleading headline and will apologise to Tony in the next issue.

Tony's email to QL Today explaining his decision in some detail was very clear and from it I could fully understand his reasons and respect him for it.

As a QL trader Tony has a distinguished record of innovation and trustworthiness that probably no other trader, past or present, can rival. It would have been remiss of me not to have paid tribute to him in QL Today."

An impression that Tony was leaving the QL community was also given by a one sentence report in the Quanta Magazine:

"The committee were saddened to hear that Tony Firshman will no longer be involved in QL activity and wish him well for the future."

QL Today understands the Quanta Magazine will also be publishing a clarification.

Important E-Mail Change

From this issue the editor will no longer be contactable by his two beeb.net email addresses, **gwicks@beeb.net** and **jstwrds@beeb.net** The beeb.net server is due to close completely on 30th June 2008.

The editor can be contacted by:

gtwicks@btinternet.com

or

QLToday@j-m-s.com

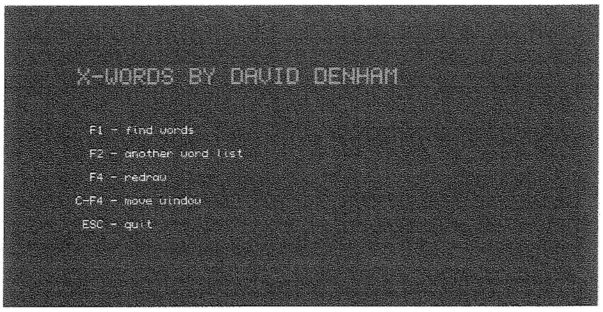
Software Section

New Program - X-WORDS

Dilwyn Jones has posted a new program by QL Today writer, David Denham, on his web page. Dilwyn writes:

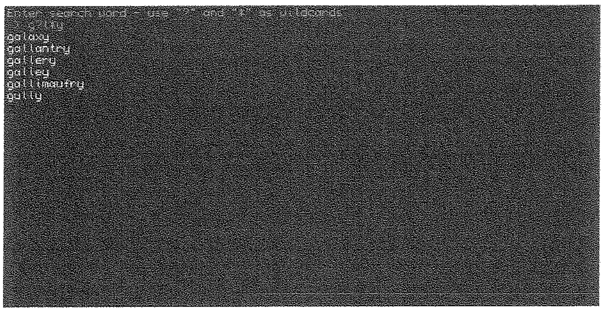
"X-Words is a simple to use crossword and word puzzle aid from David Denham. It is supplied with a word list of about 21,000 words. It can actually use any alphabetically sorted plain text word lists, there are plenty of them available, or create your own in a plain text editor, one word per line like most such word lists. The author claims to have tested it with much bigger word lists, although the 21,000 word list supplied contains many English words I've never seen before. It should be able to use other language word lists, although the author hasn't actually tested any.

It lets you use wildcards to search its dictionary, e.g. entering M?C*E would find the word MACHINE. A '?' stands for one unknown letter, while a '*' represents an unknown group of letters. Another example might be entering ????? to list all 5 letter words, or P* lists all words of any length starting with P, or *ING lists all words ending with ING.



```
X-WORDS BY DAVID DENHAM

  F1 - find words
  F2 - another word list
  F4 - redraw
C-F4 - move window
 ESC - quit
```



```
Enter search word - use '?' and '*' as wildcards
g?l*y
galaxy
gallantry
gallery
galley
gallimaufry
gully
```

*And that's all there is to it really, a nice simple to use little program which does just one specific job and nothing else. When you're as bad at solving crosswords as I am, any help like this you can get is most welcome. I guess it could probably find use as a simple spelling aid too - just enter the letters of a word you know and replace those you don't know with ? and * and it will list matching words, which you can then copy as required.*

X-Words is freeware. It is a Turbo compiled program. It is not pointer driven but seems to work OK on QPC2 and other systems with pointer environment. It needs Toolkit 2, although of course that is pretty standard on most systems nowadays."

QL Today has tested the program with both longer and foreign word lists and can confirm that such lists can be used. The program can be downloaded from:

www.dilwyn.uk6.net/games/index.html

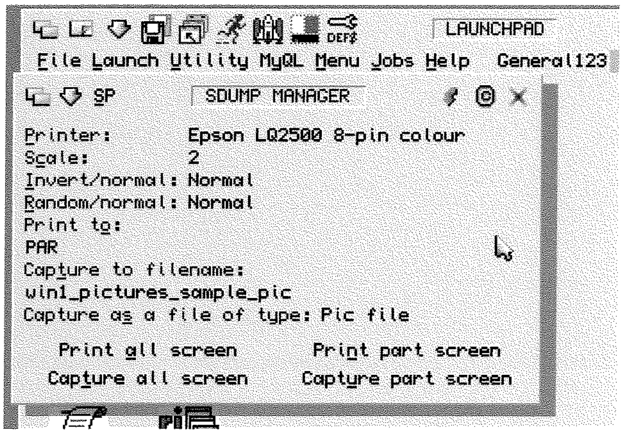
New Program - SDUMP Manager

Dilwyn Jones has also released one of his own programs:

"This is a new pointer driven screen capture program plus an interface to act as a front end for the SDUMP screen dump extensions built into some disk interfaces and supplied as an extensions file with SMSQ/E, although current versions of SDUMP only work in mode 4 and mode 8. Capture or print all or part of a screen. Screen capture works in all modes and can save to screen files, Pic files, Psa files, 24-bit BMP files (for transfer to a PC) and mode 4 Page Designer 2 page files."

The program can be downloaded from

www.dilwyn.uk6.net/graphics/index.html

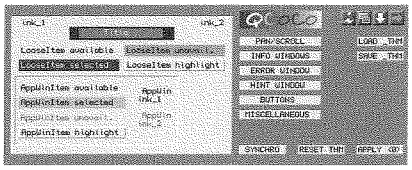


```
Printer:        Epson LQ2500 8-pin colour
Scale:          2
Invert/normal:  Normal
Random/normal:  Normal
Print to:
PAR
Capture to filename:
win1_pictures_sample_pic
Capture as a file of type: Pic file
```

Upgrade - QCOCO

Bob Spelten has updated QCoCo to version 1.60. The main changes are:

- To make it more usable on Aurora, I changed the background images. They are no longer taken from _bmp files but are now loaded from big sprite files. Consequently the "qlborder" image is now also a sprite and part of the program code, so it doesn't need to be loaded from the root anymore.
- When changing item attributes, the current item number and the colour number are shown in the selection menu.
- The Favourite colours can now be loaded and saved from the selection menu, therefore a new extension is preferred: "_mfc".
- Colours can be selected from Favourites as before but now also from the QL colours, the PAL colours, the System colours (which are the actual colours in the current theme), the Grey shades and even PAL stipples can be made. With the '‹' or '›' buttons you can scroll through all the colours.
- New colours selected for an item will be added to the System colours and can thus quickly be picked from there for other items. If on reflection a colour change wasn't the right one, you can always pick the old colour back again from the System colours.
- "Load theme" can now read the colours from any of the 4 system palettes while "Apply" can save the current theme to any of the 4 system palettes. Doing a "Load theme" or "Apply" will also change the system palette the program is using itself. A new "Reset" button will undo all changes since the last save or load and redraw the main window.
- Minor bugs have been fixed and the menus have been updated in EasyPtr4.
- All menus have an OK option to confirm the colour changes. Escape will undo the most recent changes for that menu and all menus have an escape option.
- All 57 system items can now be changed individually. Hitting the space above the Title item will list them all.
- When you Hit the QCoCo name in the Info menu, more program info is available and you can change the background colour or skin sprite or override some configurated defaults. 5 more defaults can be set with (Menu)Config.
- Some Menu_rext menu calls can already use the new 'timeout' options of version 7.67 (still under construction).

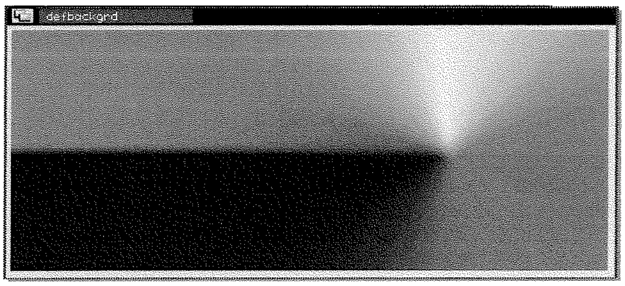


Upgrade - Sprite-View/Menu-View

Bob has also upgraded Sprite-View and Menu-View to version 1.01

- Some error trapping has been added so they don't crash anymore if the sprite or menu is too big.
- Sprite-View has a scalable window and can now show sprites almost as big as your screen. Small sprites will be shown as pointer sprites as before, while big sprites fill the View window which also has a Move item and shows the sprite name.
- Hitting F3 in the Menu-View window will write the current menu statistics to a file in ram1, any other key will quit.
- When you execute these programs without a command string, they will open a file select window and each time you click on the window a new filename can be selected. But you can also configure FileInfo2 to call them on the _spr or _men extension as before. Then a click on the window will quit the job.

The upgrades can be downloaded from:
http://members.upc.nl/b.spelten/ql/



Maps CD Released

Just Words! has now released its CD of mapping data for use with the mapping articles by Hugh Rooms currently being published by QL Today.

The CD contains latitude and longitude data for each of the five continents in two versions, one with rivers and one without. These files are too large to be placed on the Just Words! website.

The CD also has data for individual countries, but these can also be downloaded from the website.

The CD costs just £1. This could be paid in postage stamps or International Reply Coupons.

Geoff Wicks,
5b Wordsworth Avenue,
Derby,
DE24 9HQ,
UK.
Telephone: +44(0)1332 - 271366
E-Mail: **gtwicks@btinternet.com**
Web: **http://members.lycos.co.uk/geoffwicks/justwords.htm**

Hardware Section

QUBIDE Board

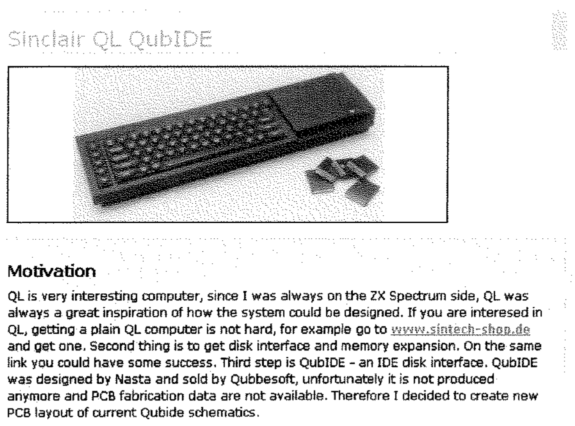
Jan Palenicek has announced he is working on a new layout for QubIDE. In an mailing to the QL-User group he wrote:

"I am wondering if anyone here would be interested to get unassembled PCBs and make Qubide on his own. Complexity is low, perhaps the biggest issue would be to get Eprom and GALs programmed.

Apart from this, I would like to know what are your experiences with Compact Flash cards and Qubide. Can you give me advice if all cards are working properly? Are there any HW changes necessary for CF cards? AFAIK there are some issues with new CF cards that are using 3V TTL technology, is this also issue on Qubide?

New PCB layout is 90% finished and it is based on the original schematics. Before we go into PCB production, is there anything you are aware of that should changed or added on the new QubIDE?"

He later added that he had finalised layout, schematics and was about to order prototype PCBs. You can follow progress on the website:
http://omega.webnode.com



Sinclair QL QubIDE

Motivation

QL is very interesting computer, since I was always on the ZX Spectrum side, QL was always a great inspiration of how the system could be designed. If you are interesed in QL, getting a plain QL computer is not hard, for example go to www.sintech-shop.de and get one. Second thing is to get disk interface and memory expansion. On the same link you could have some success. Third step is QubIDE - an IDE disk interface. QubIDE was designed by Nasta and sold by Qubbesoft, unfortunately it is not produced anymore and PCB fabrication data are not available. Therefore I decided to create new PCB layout of current Qubide schematics.

Which Monitor?

Arising from Jan's news of a new layout for QubIDE, **Peter Graf** raised an interesting question on the QL-users group:

"One thing puzzles me: You folks still use the black QL or (S)GC, so how do you get a display attached?

I have a TFT monitor, and barely get my Q60 to work by a special video chip. Black bars on top & bottom, not a nice solution. Found no way at all to attach my Super Gold Card.

I was even thinking about making a QL video card for myself. But I wonder why there seems *to be no need for such a thing. At least nobody complains in public. Do you help yourself with Aurora or still use old QL monitors or do you use some other video converter?"*

Quanta's 2004 survey of QL users showed about 83% still had some form of native hardware and 34% cased QLs. However little is known of the monitors used with native hardware and Peter's question produced little response. Two people replied that they used a Commodore 1084S. The editor believes this was a Phillips model that was also sold under other brand names and was one of the most popular monitors used by QL-ers.

Scarcely anything has been published on the feasibility of an interface to use modern monitors with native QL hardware. This would make an interesting topic for a short article by one of the hardware experts.

Cut-Price QUANTA

Customers of **RWAP Software** who are not members of Quanta can join for a trial period of 1 year for just £7, half the normal subscription price. A condition of this offer is that they agree to opt for the electronic version of the Quanta Magazine.

Head of RWAP Software, Rich Mellor, has forged close links with Quanta in recent years by selling software on eBay on their behalf. Rich's most successful lines are Spectrum and QL keyboard membranes, and the last batch of these was financed by a loan of £2,150 from Quanta.

In the last five years Rich has bucked the general trader trends by increasing his turnover and even making a profit. Elsewhere in this issue you will find a short article, commissioned by QL Today, in which he describes how he has achieved this.

Website Restored

Norman Dunbar writes:

"The web site formerly known as 'Qdos Internals' has been off line for some time but now it is (almost) back again!

I have apparently got 5 sub-domains at dunbar-it.co.uk so I've paid a hosting company the massive sum of £2 per month (or £22 per year) to give me 125 MB and a decent bandwidth, plus I can just about install anything I want with very few restriction. (www.5quidhost.co.uk) They have a free option too - pay them a visit, you might be pleasantly surprised.

Anyway, I've created a Wiki on my sub-domain and I'm in the process of converting my old HTML pages into a Wiki format. Once I'm done with the conversion, I'll be opening the Wiki up to all and sundry (well, to subscribers only) so that we can collectively create the definitive online QDOS SMSQ/E repository of knowledge! (or something like that anyway).

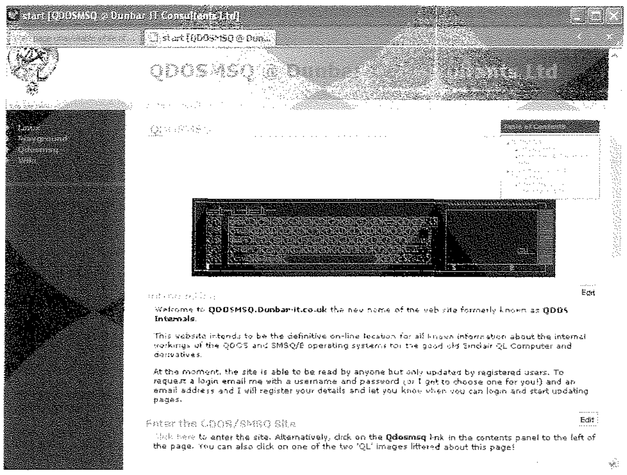
The web site is

www.qdosmsq.dunbar-it.co.uk

Remember it is under construction but I find a Wiki easier to create and change as I'm behind a huge firewall at work, so lunch time updates as in the past, are no longer possible as FTP etc is restricted.

Maybe one day I'll actually finish it!"

As the name implies the site will be of most interest to programmers. The site currently has four main areas - Linux, Playground, QDOSMSQ and Wiki. The Linux section includes details on reading QL formatted files under Linux. The playground is still empty and the QDOSMSQ section the most extensive. This contains sections on the QL filing system, jobs, memory, SuperBasic internals, system variables and traps.


start [QDOSMSQ @ Dunbar IT Consultants Ltd]
start [QDOSMSQ @ Dun...
QDOSMSQ @ Dunbar IT Consultants Ltd
Linux
Playground
Qdosmsq
Wiki
Edit
Edit

Welcome to **QDOSMSQ.Dunbar-it.co.uk** the new home of the web site formerly known as **QDOS Internals**.

This website intends to be the definitive on-line location for all known information about the internal workings of the QDOS and SMSQ/E operating systems for the good old Sinclair QL Computer and derivatives.

At the moment, the site is able to be read by anyone but only updated by registered users. To request a login email me with a username and password (or I get to choose one for you!) and an email address and I will register your details and let you know when you can login and start updating pages.

Enter the QDOS/SMSQ Site

Click here to enter the site. Alternatively, click on the **Qdosmsq** link in the contents panel to the left of the page. You can also click on one of the two 'QL' images littered about this page!

ASUS eeePC

In the last issue of QL Today Roy Wood wrote about the Asus EEEPC a small laptop computer priced at just over £200. Measuring just 225mm x 170mm and weighing in at 1.12Kg it has a 7in (178mm) 800x480 display, no hard drive, CD or floppy but instead flash memory of up to 8Gb. At the time Roy wrote about the machine it had a Linux operating system and thus was unsuitable for QPC2.

Since then a Windows XP model has become available, and there has been some discussion on the QL-user email group about its suitability as a portable QL running QPC2.

Tobias Fröschle described his experience:

"Nice little beast.

Mine is running uqlx and linux, at the moment, so I have no experience with QPC2 on it - uqlx runs fine, though. What you should have is:

Tiny fingers and sharp eyesight.

as both keyboard and screen are really, really small. uqlx at 800x480 gives on-screen characters 3-4 mm high - So you should be used to read the small print. If you are a 10-finger touch-typist, forget the eeePC - you won't fit them on the keyboard all together.

It's fast enough for serious work and small enough to carry it everywhere. Got 3 free USB ports, so easy to expand. But you need to get used to the small scale."

Later Tobias installed QPC2 and reported he could see little difference between the eeePC and a "big" laptop. Dr. M. Hamias reported a similar experience.

Jimmy Montesinos wrote that he had been surprised by its performance and that QPC2 works well in both Windows and full screen mode. However he would not recommend the machine for daily use, but "it is just fun to use".

Jimmy also announced modifications to QL2K to make better use of the 800x480 screen. (He also mentioned the next release of QL2K will have sound emulation.)

On page 42 is a full review of the Asus eeePC.

Mapping Ancient and Modern – Part 2

by Geoff Wicks

Last time we looked at some of the traditional QL programs for drawing maps. The big problem with these programs was collecting the map drawing data, which meant hours of work with a ruler or tracing graph paper. The good news about Hugh Room's program is that the latitude and longitude data already exists on the internet. However downloading and editing this data is not easy and it will stretch your hardware and software to the limits.

When Hugh sent his draft suggestions for a series of mapping articles, one of my first queries was how easy it would be for readers to try out the programs. His initial answers were not encouraging because the data bases he was using were extremely long. For example, it would take two or three complete issues of QL Today just to print out the data base for the British Isles.

I asked Hugh to point me in the right direction to search the internet for data bases, and he hit the jackpot with his first suggestion. This was the site:

http://www.evl.uic.edu/pape/data/WDB

WDB stands for "World Data Bank" and is a data base originally compiled by the CIA in the 1980s. It has since been modified to a text format that can be downloaded from the site. There are five separate files giving data for each of the continents varying in size from 3.1Mb for Europe to 10.6Mb for Asia. This is the compressed size and the compression is not in the usual zip format but in tar format, which I believe is a compression program used more in the Linux than the PC world.

The uncompressed Europe data base is 11.7Mb and that for Asia is 40.4Mb. You have to modify the files to the format Hugh Rooms uses. This is latitude - space - longitude. The WDB files contain a lot of explanatory text as well as the latitude and longitude data, which is in the format tab - latitude - tab - longitude.

Removing the explanatory text is much more difficult than you might expect. As a text file this article on mapping is smaller than 8Kb. Expanded Europe, the smallest of the WDB files, is almost 1,500 times larger - the equivalent of about 50 issues of QL Today.

Way back in 2003 I wrote an article for QL Today entitled "Going to Extremes" (Vol. 7. issue 5 page 16) describing how editing word lists for dictionaries stretched your text editor or word processor to its limits. The maximum file size of the word lists I was using was just over 2.5Mb. The WDB data bases are 4 to 16 times larger.

On the PC Lotus WordPro and Notepad could not load the data bases although Wordpad could. I do not possess Microsoft Word to try but Word Viewer could load the files. The one file I tried editing in Wordpad became corrupt, but this might have been coincidence.

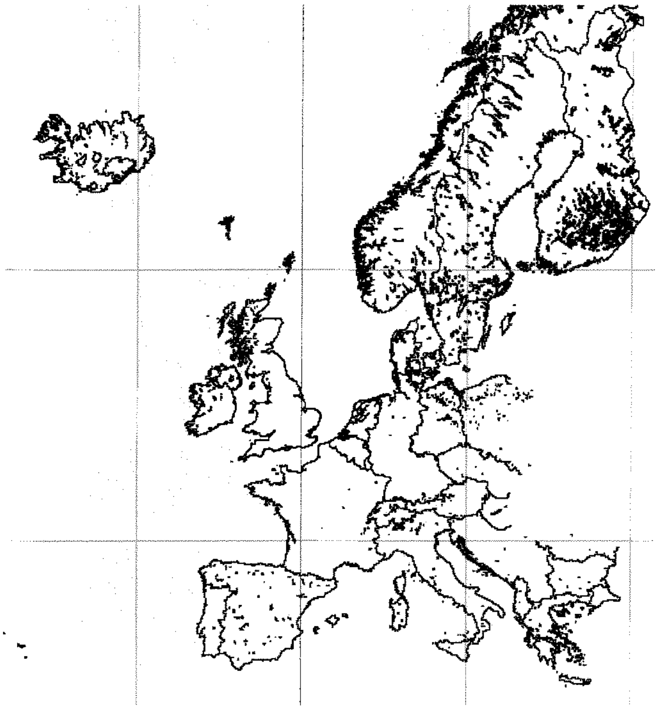
On the QL most text editors and word processors have memory restrictions that make them unusable. I was forced to fall back on Perfection, which seems to be able to cope with practically anything you throw at it provided you have enough memory. And don't forget you have to have not only sufficient memory to hold the data base, but also memory for the text editor or word processor to do its work. I never thought I would see the day when I would have to configure QPC to its maximum memory size of 128Mb.

Another problem is that the WDB data is arranged in three sections. The first is national boundaries; the second coastlines, islands and lakes; and the third rivers. I found it better to omit the last section as otherwise the maps can become too cluttered.

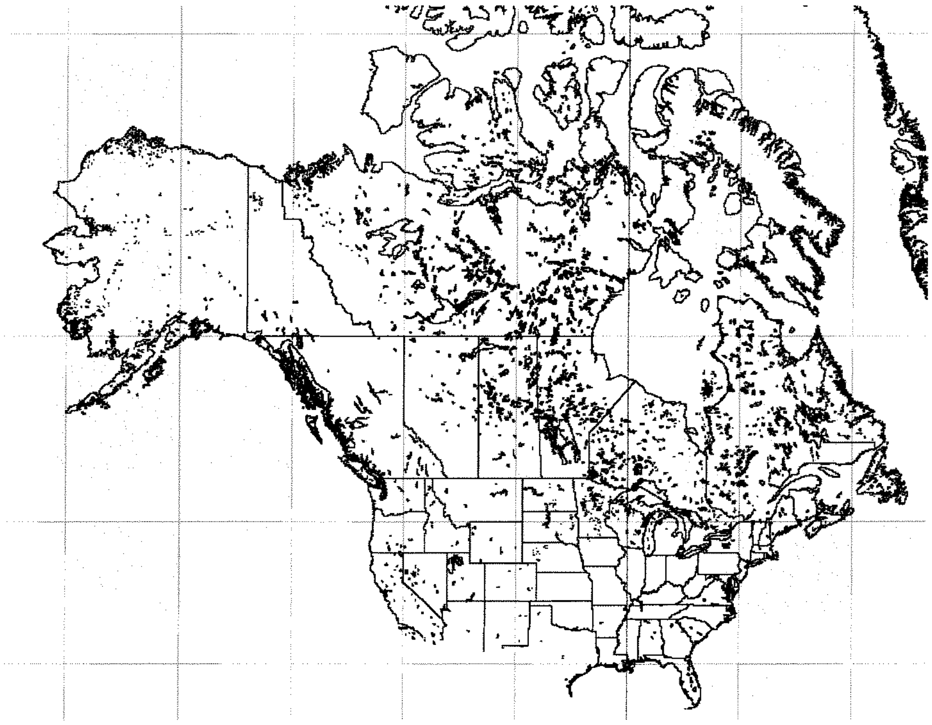
Once you have got rid of the extraneous matter, you then have to modify the data into Hugh's format. This is a fairly simple matter involving a few lines of SuperBasic. After modification and the removal of the river data the files were still huge. Europe was over 7Mb containing over three quarters of a million items of data and Asia over 20Mb containing over 2 million items of data.

Fortunately SuperBasic again comes to our rescue because it is fairly easy to extract the data for an individual country provided you know the maximum and minimum latitude and longitude. Great Britain is 943Kb and 97,000 data items; Benelux 245Kb and 26,000 data items; Italy 919Kb and 95,000 data items; and Japan 763Kb and 74,000 data items.

Why do the files have to be this big? One reason is that you cannot use the SuperBasic LINE command to draw the maps because there is no way of knowing, in the UK for example, when you are drawing the mainland and when an island. You have to build up a map by a series of dots using the POINT command.



Europe



North America

Just Words! can save you the bother of downloading and modifying the data bases, because we have extracted the data for several of the countries where QL Today readers live. We hope to put these files on our website so that you can try out Hugh's program straight away. At the time of writing the Just Words! maps webpage still has to be written and tested, but the latest information should be in QL Today's news pages. If not, go to the website to see if there is a link to a maps page:

http://members.lycos.co.uk/geoffwicks/justwords.htm

Unfortunately we cannot put the full continental databases on the website as this would exceed our bandwidth. However we hope to produce a CD containing this information for a nominal fee. We also hope to introduce a "tailor made" service to provide the mapping data for any part of the world where you provide us with the maximum and minimum latitude and longitude.

If we now take a look at Hugh's SuperBasic program, the first thing to note is in the first line. Hugh has called his program Qtest_bas. That is an accurate description. It is not a finished program, but simply does the job of testing his theory. The program is there for you to adapt for your own system. Indeed Hugh himself points out in lines 180 and 190 that you may have to do this. He assumes you have a screen larger than 995 x 670 pixels and you may have to change this. Equally in line 150 he assumes you have GD2 colours. Another obvious change is the name of the input file in line 350.

To use his program to the full you will need to be fully conversant with the SCALE command. In line 270 is his scaling for Great Britain and in 320 the scaling for a world map. If you use any of the Just Words! data bases you will need to adjust the scaling.

The syntax of the command is:

```
SCALE [#channel,] height, x, y
```

where x and y are the graphical co-ordinates of the bottom left hand corner of the window.

In very simple terms the smaller the height the larger the image. Increasing the x co-ordinate moves the image to the left, decreasing it to the right. Increasing the y co-ordinate moves the image downwards, decreasing it upwards. And don't forget that both co-ordinates can have negative values.

If you are uncertain the best advice is to change the height in small stages and then adjust the x and y co-ordinates to keep the image in the centre of the screen.

To help you on your way, here is the scaling you can try with the Just Words! data bases:

Australia and New Zealand: 50,70,-55
Austria and Switzerland: 8,4,50
Benelux 10,-5,55
British Isles: 20,-10,57
Canada: 120,-120,40
France 18,-10,45
Germany: 15,0,53
Greece: 10,10,38
Iceland: 10,-20,82
Italy: 15,0,39
Japan: 28,80,30
Scandinavia: 46,-30,60
Spain and Portugal: 15,-15,35
USA: 60,-100,10

Finally a couple of programming challenges.

First would anyone like to try writing a routine that would automatically adjust the scaling for the optimal reproduction of a map, given any maximum and minimum latitude and longitude?

Secondly who would like to try incorporating Hugh's routines in his own program? There are two big problems to be overcome. One is the size of the data base and the other the speed of writing. However you could investigate the possibility of reducing the size of the data base by taking say only every fifth or tenth data pairs and testing to see how this would affect the quality of the map.

Happy mapping!

RAMDiscs

by David Denham

We are all used to the fact that commands using the QL devices work in pretty much the same way, no matter what the device name may be – we use the same commands to save to floppy disc, microdrive, romdisq and hard disc, with only the device name changing.

For example, to save a SuperBASIC program to microdrive, we use the command

```
SAVE mdv1_progname
```

To save it to floppy disc, we just change the mdv to flp:

```
SAVE flp1_progname
```

And to save to Tony Firshman's Romdisq device, we use:

```
SAVE rom1_progname
```

The only difference as far as you are concerned is the drive name – this is called ***device independence***.

The operating system itself has to work a little differently with all devices – as you might expect, microdrives are tapes with a capacity of 100 kilobytes whereas a floppy disc uses a magnetic disc with capacity of anything from 360 kilobytes up to 3.2 megabytes. Each device may work at different speeds too – a hard disc may be much faster than a microdrive cartridge for example. A modern hard disc can be huge too. All have different electronics controlling them, and so they all have their own "driver" which is a chunk of machine code which knows how to "drive" a particular device (hence the name "driver"). The operating system tries to hide all this from you so that as far as you are concerned the only things which are different from device to device is the name and how much data it will store.

One device which doesn't get mentioned very often is the ramdisc. This is similar to a floppy disc in some ways, but exists only in memory, so its content is lost when the QL is switched off or reset. So what use is this if the content is lost when switched off?

The answer is that its main use is as a fast temporary storage device. A simple example is when your QL has only a single floppy disc drive. How do you copy or backup a floppy disc on that system? You could copy files to a microdrive cartridge of course, but if the disc is full, the content may not fit onto a cartridge, and it may take a long time since microdrives are usually slower than a floppy disc. This is where a ramdisc comes in – the ramdisc can hold as much as your computer's memory allows and it is fast. So you can copy all the files from the floppy disc to a ramdisc quickly, then insert a new floppy disc and copy all the files from ramdisc to this disc.

There are two main types of ramdisc – the ***Static*** (fixed size) type and the ***Dynamic*** (variable size) type.

A ***static ramdisc*** is a fixed size one, using whatever capacity you assign to it using a FORMAT command, whereas a dynamic ramdisc can grow in size or shrink depending on the size of the files it stores. All this resizing is automatic and it usually works well unless it leads to common heap fragmentation if the resizing is alternated frequently with other programs reserving and releasing chunks of common heap memory.

Fixed size ramdiscs are set up using a FORMAT command with the ram drive name and number followed by the number of 512 byte sectors it is to have, e.g. FORMAT ram1_220 would create a ramdisc drive number 1 with a capacity of 220 sectors, roughly equivalent to the typical capacity of a microdrive cartridge. A dynamic ramdisc does not need a format command to start it – just use it. Dynamic ramdiscs start with a capacity of 0 and expands and contracts automatically to meet the needs of the time.

Either type can be reset to zero capacity with a format command using the ram drive name and number with a capacity value of 0 or just the drive name and number:

```
FORMAT RAM1_0
```

or just:

```
FORMAT RAM1_
```

Most ramdisc systems support up to 8 drives (RAM1_ to RAM8_), although some older types may only support 1 or 2 drives.

Most fairly modern disc interfaces, including all disc interfaces from Miracle Systems, include drivers for ramdiscs. Some of the earliest disc interfaces do not, and neither do some QL emulators such as Qlay. Luckily, there's public domain ramdisc software out there to let you use either static or dynamic ramdiscs – you can get such software from Dilwyn Jones's website:
http://www.dilwyn.uk6.net/tk/index.html

With that software, you can either LRESPR the ramdisc driver code or run the relevant boot program.

Sometimes we need to decide whether to use a static or dynamic ramdisc, and the answer is not always easy to arrive at. If you need the flexibility of a drive which can change its size automatically as and when the need arises, you need a dynamic ramdisc. Conversely, if you have a need for setting up ram drives of the same capacity as a microdrive cartridge or floppy disc, use a static ramdisc formatted to the capacity required. Static ramdiscs are also useful if other programs on the system are allocating and releasing memory all the time – assigning a fixed size can help reduce potential heap fragmentation problems. Actually, most dynamic ramdiscs can be formatted into fixed capacity ones so that they can function as a static type if required.

Some ramdisc systems even let you change the driver name. This can be useful if you wish to run a program configured to run from microdrive cartridge and the system concerned (e.g. an emulator) has no microdrives. In this case, copy the microdrive program to a QL floppy disc on a real QL system, then when you want to run it on a system with no microdrives, copy all the files from the floppy disc to a ram drive and rename the RAM driver name with a RAM_USE command to change the three letters of the drive name:

```
RAM_USE "MDV"
```

This changes the device name from RAM to MDV, so that when you do DIR mdv1_ for example, what you actually get is a list of files on RAM1_. The program should then run from ramdisc, thinking it is running on a microdrive cartridge. This is called ***Microdrive Emulation***. Note: older ramdiscs may not support RAM_USE.

There is another unique and useful function of ramdiscs: microdrive imaging. Again, older ramdisc software may not support this facility. Normally, it can take ages to copy all files from a microdrive cartridge to another medium, especially if there is any sort of problem with cartridge readability. Microdrive imaging means making a direct copy of everything on the cartridge into the ramdisc, which should not take more than 2 complete revolutions of the cartridge tape before all imaging is finished (typically 15 seconds).

To use microdrive imaging, a special form of the FORMAT command is used, where the ramdisc drive and number are followed by the name and number of the microdrive to copy into ramdisc. Here's an example where the complete content of MDV2_ is copied to RAM1_

```
FORMAT RAM1_MDV2_
```

Once the files have been copied and the microdrive has stopped, you can check that the copy has been successful with a DIR command:

```
DIR RAM1_
```

Which should give a normal-looking list of files. Watch out for any files with an asterisk appended to the name, which means that at least one sector of that file had some form of problem and was not copied properly, so the program or file may be corrupt or unreliable.

Assuming all files were copied without a problem, you can then use those files in any way you need, e.g. copy them to disc.

Other Uses for Ramdiscs

We've already discussed using ramdiscs for copying files and for microdrive emulation and imaging. Here, we'll discuss other uses of the ramdisc.

Archive Databases

I'm sure we've all experienced an Archive database which refuses to load, usually after we forgot to close it properly, or suffered a power cut before the file saving was complete. Archive can also be rather slow when doing some things, especially with larger databases. One solution is to use the database while in a ramdisc and copy it to floppy disc or hard disc once you have finished using archive. Suppose we have a database called club_dbf. First, we copy it to ramdisc:

```
COPY FLP1_club_dbf TO ram1_club_dbf
```

Then, in Archive, use the OPEN command to open it for read and write. Then do any necessary updates, close the file and quit from Archive. Finally, copy the updated database to disc with the COPY command:

```
COPY ram1_club_dbf TO flp1_CLUB_dbf
```

One problem you may encounter when using Quill or Archive in this way is that these programs may grab all available free memory, leaving no room for the database to grow, as the dynamic ramdisc cannot expand when there is no free memory. This is a good example of a case where a static ramdisc may be best, since we can fix the size of the ramdisc before Archive or Quill grab all the free memory, so we can use a format command to set the size of a ramdisc to be the same as the disc (assuming our system has enough memory), and write a short SuperBASIC program to control and partially automate all this. It assumes that Archive is in FLP1_ and the CLUB_DBF file is on FLP2_

```
100 REMark fix size of ramdisc as DD disc
110 FORMAT ram1_1440
120 REMark transfer a working copy of
    database
130 COPY FLP2_CLUB_DBF TO RAM1_CLUB_DBF
140 EXEC_W FLP1_ARCHIVE
150 REMark copy database back to floppy disc
160 COPY RAM1_CLUB_DBF TO FLP2_CLUB_DBF
```

Another use for ramdiscs is, strangely enough, for holding program help files so that they do not have to be loaded from disc or cartridge when needed. In this case, the program concerned needs to be configured so that it looks for its help files on the ramdisc. In this way, the help files are displayed quickly on request, provided of course that they have already been copied there (e.g. by a boot program).

A Tick Timer - Part 2

by George Gwilt

In Part 1 I described the stages in the development of the program "timer1_asm".Its source code is given below with comments, but it might be useful if I make a few more general remarks here.

Unlike Simon Goodwin who had four stopwatches in his timing system, I have only one timer. However you can use my system to time any number of events. For each one you record the value of the timer at the start. This is equivalent to starting a stopwatch. When you want to find how much time has elapsed you find the current value of the timer and subtract that from the initial, recorded, value. This is equivalent to reading the stopwatch.

You might wonder if it is wise to leave the timer going even when you did not want to use it. Turning it off would save time at each tick. However, since the timer requires only two instructions to be linked into the polling list the extra time taken is tiny.

You might also wonder if the timer might overflow. In fact it would have to run continuously for over two and a half years before that.

Finally, I might remind you that the config block which appears nearly at the end of the listing, was produced by the program UCONFIG (for universal config). You might be surprised that at its start are four lines of explanation between "/*" and "*/". These are the codes indicating the start and finish of comments in a C program. They also act as indicators to GWASS that the lines are comments. In fact the same heading is used by UCONFIG for the output of both assembler and C config blocks.

Source Code

```
; timer2_asm
;
; EX timer2_bin;par$ will set a tick timer in SYS_VARS + $DC
;
; The optional parameter string par$ must be -wx where x=0 or 1
;       x=0 for the result set in a window
;       x=1 for the result returned as an error code (or 0)
; The configurable item governing this by default is the byte at CDEA0
;
          in        win1_lib_hed1
          bra       start
          dc.l      0
          dc.w      $4afb
          hed1      <"Timer">,fname

sys_tim   equ       $dc

*************************************
*                                   *
*       Process parameter list      *
*                                   *
*************************************
```

The optional parameter list, to have any effect, must be either "-w0" or "-w1".

The purpose of this is to set up the choice between ending with a message in a window or ending with an error code. There is a config block at the end of the program which contains the one configurable item. It is a code called CDEA0 having the value 0 or 1. This value is set in D5.B.

Since the parameter list may override this code it must be processed now. To be able to examine the list a byte at a time, we copy the address of the stack to A0 since an odd address of the stack is not allowed.

If there is a parameter it must be of the correct form. If not it is ignored. If it is of the correct form then 0 or 1 is put in D5.B.

```
start     movea.l   sp,a0               keep stack address in A0
          move.b    cdea0,d5            0=windows 1=error return
          move.w    (a0)+,d0
          lsl.w     #2,d0               no of channels*4
          lea       (a0,d0.w),a0        point to parameter string
          move.w    (a0)+,d0            length of par string
          beq       start1              no string
          subq.w    #3,d0
          bne       start1              not a correct string
          move.w    (a0)+,d0
          cmpi.w    #"-w",d0            a correct parameter? . .
          bne       start1              . . no
          move.b    (a0),d0             "0" or "1" expected
          subi.b    #'0',d0             integer 0 or 1 expected
          bmi       start1              < 0 !!
```

```
        cmpi.b  #1,d0
        bgt     start1          › 1 !!
        move.b  d0,d5           set windows/error marker
start1

****************************************
*                                      *
*             Clear stack              *
*                                      *
****************************************
```

This is not strictly necessary, but it always seems to me that we might as well start with as big a stack as possible. If someone starts a program with a parameter list 32,000 bytes long, which is always possible, these extra bytes will not be available unless we clear the stack. This is done by setting the stack equal to A6.L plus A5.L.

Of course we must remember not to do this before we have processed the parameters since you can never rely on previous contents of the stack remaining. This is because a program's stack might be used by the scheduler while the program is temporarily in abeyance. Such use will cover up previous contents. For this reason the clearing of the stack is done after the parameters are processed.

```
        lea     (a6,a5.l),sp    clear stack

****************************************
*                                      *
*         Check if timer set           *
*                                      *
****************************************
```

The timing count will be zero if the timer is not set. It is possible that someone else is using sys_tim and has set 0 there. I take this as very unlikely. Hence a zero count leads to the timer having to be set. If the count is not zero, either the timer has already been set, or someone else is using sys_tim. To check this I make the program wait three ticks (one should be enough I suppose) and check again. If there is a change I assume that this is due to the timer.

The results of these tests show whether to issue a message straight away or to set the timer.

Note that the code defining the message is set in D7.L.

```
          moveq   #mt_inf,d0
          trap    #1
          move.l  sys_tim(a0),d7    time
          beq     set_timer         timer not set
          moveq   #3,d3             wait 3 ticks
          bsr     wait
          cmp.l   sys_tim(a0),d7    altered? . .
          beq     cant_set  . . no
          moveq   #2,d7             message 1 - already set
          bra     do_mess
cant_set  moveq   #4,d7             message 2 - in use
          bra     do_mess
no_space  moveq   #6,d7             message 3 - out of memory
          bra     do_mess

****************************************
*                                      *
*         Set timer routine            *
*                                      *
****************************************
```

We need to allocate space for the link. The amount of space is eight bytes for the link itself plus the length of the routine to be linked. Although I know that the routine is six bytes long, the program is set to calculate this from labels set at the start and finish of the routine. Thus we see

```
          #rout_e-rout+8
```

instead of

```
          #14
```

The point of this is that a generalisation of the routine at "rout" will not then need a change in the instruction filling D1.L. Unless, of course, the routine takes more than 126 bytes!

If space cannot be allocated, we branch to "no_space" to record this. Of course we will probably not be able to open a window to record the sorry lack of space because of – lack of space. However, in this case we should eventually end up with an error message being sent instead.

If we successfully acquire space, owned you should note by master BASIC, we fill that with the routine itself as well as its address.

This is now linked into the polling list by MT_LPOLL.

A successful link results in D7.L being zeroed to indicate the success.

```
set_timer addq.l    #1,sys_tim(a0)      set time non zero
          moveq     #rout_e-rout+8,d1   bytes needed for link
          moveq     #0,d2               BASIC owner
          moveq     #mt_alchp,d0
          trap      #1                  get space
          tst.l     d0
          bne       no_space
          lea       8(a0),a1            place for routine rout
          move.l    a1,4(a0)            set this address in the link
          lea       rout,a2
          moveq     #rout_e-rout,d0     length of routine
          bra       lp2
lp1       move.b    (a2)+,(a1)+         rout to linkage block
lp2       dbf       d0,lp1
          moveq     #mt_lpoll,d0
          trap      #1
          moveq     #0,d7               message 0 - timer set

**************************************
*                                    *
* Test if "window" or "error message" *
*                                    *
**************************************
```

At this stage we are about to finish. The error message code is in D7.L and the code for the method of its delivery is in D5.B.

```
;
; At do_mess we send results to a window if D5.B = 0 else "error" return
;
do_mess   tst.b     d5
          bne       do_er

**************************************
*                                    *
*                window              *
*                                    *
**************************************
```

The code indicates "window" so we open a window by the subroutine "ope".

The messages to send are listed just above the table tble0. These use the macro "hed1" given after the program's listing. This sets up the instructions for a string equal to the first parameter, which is the contents of ‹ .. ›This is given a label equal to the second parameter.

The table in tble0 is in the common form of a list of word offsets from the start of the table to the required heading. The offset we want is obtained by using an index, in this case D7.W which contains an even number between 0 and 6.

D7.W also acts as an index into table tble1 which contains the x-values for the positioning of the cursor for printing the heading.

```
          bsr       ope                 open a window
          lea       tble0,a1
          lea       tble1,a4
          move.w    (a4,d7.w),d1
          moveq     #2,d2
          bsr       at
          adda.w    (a1,d7.w),a1
          bsr       utm
          move.w    #200,d3
          bsr       wait

***************************************
*                                     *
*           Exit from program         *
*                                     *
***************************************
```

The usual way for a program to exit is to forcibly remove itself. The contents of D3.L will be returned to the calling program in D0.L if it is waiting.

```
sui       moveq     #0,d3
sui1      moveq     #-1,d1
          moveq     #mt_frjob,d0
          trap      #1

***************************************
*                                     *
*                 Tables              *
*                                     *
***************************************

          hed1      ‹"con_128x64a20x10"›,con
          hed1      ‹"sys_tim set now"›,h0
          hed1      ‹"sys_tim already set"›,h1
          hed1      ‹"Can't set timer",10,"   sys_tim in use"›,h2
          hed1      ‹"Can't set timer",10,"    Out of space"›,h3
          hed1      ‹"George Gwilt's Timer"›,h4
          hed1      ‹"sys_tim = $DC"›,h5

tble0     dc.w      h0-tble0
          dc.w      h1-tble0
          dc.w      h2-tble0
          dc.w      h3-tble0

tble1     dc.w      2,0,2,2

***************************************
*                                     *
*              Subroutines            *
*                                     *
***************************************
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                     @
@            Opens a window           @
@                                     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

ope       lea       con,a0
          moveq     #-1,d1
          moveq     #0,d3
          moveq     #io_open,d0
          trap      #2
          moveq     #-1,d1              colour grey
          moveq     #2,d2               border width
          moveq     #-1,d3              timeout
          bsr       border
          moveq     #2,d1               red
          bsr       paper
          moveq     #0,d1               black
          bsr       ink
          bsr       cls
          lea       h4,a1
          bsr       utm                 prints heading h4
          moveq     #3,d1
          moveq     #5,d2
          bsr       at                  AT 5,3
          lea       h5,a1
          bsr       utm                 prints heading h5
          moveq     #7,d1               white
          bsr       ink
          rts

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                     @
@       Waits for D3.W ticks          @
@                                     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

reg_wt    reg       d1/a0
wait      movem.l   reg_wt,-(sp)
          moveq     #mt_susjb,d0
          suba.l    a1,a1
          moveq     #-1,d1              this job
          trap      #1
          movem.l   (sp)+,reg_wt
          rts

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                     @
@   AT, CLS, BORDER, INK and PAPER    @
@                                     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

at        moveq     #sd_pos,d0
at1       trap      #3
          tst.l     d0
          beq       at2                 OK
          addq.l    #4,sp               prepare to bow out
          bra       do_er               no window so do error message
at2       rts
```

```
cls       moveq     #sd_clear,d0
          bra       at1

border    moveq     #sd_bordr,d0
          bra       at1

ink       moveq     #sd_setin,d0
          bra       at1

paper     moveq     #sd_setpa,d0
          trap      #3
          moveq     #sd_setst,d0
          bra       at1

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                     @
@           Print string @ A1         @
@                                     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

utm       movea.w   ut_mtext,a2
          jmp       (a2)
;
;  we return an error message via D3
;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                     @
@       Return error code in D3.W     @
@                                     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

do_er     lea       ertable,a0
          move.w    (a0,d7.w),d3
          ext.l     d3
          bra       sui1

ertable   dc.w      0,-1,-9,-3

***************************************
*                                     *
*       The actual timer itself!      *
*                                     *
***************************************


;
; Routine to be linked to poll list
;
rout      addq.l    #1,sys_tim(a6)      add 1 to time in ticks
          rts
rout_e    ds.w      0

***************************************
*                                     *
*           The config block          *
*                                     *
***************************************

/* Configured Items - Config Level 1
 *
 * Name     Description
 * CDEA0    Window or Error return
 */

  DC.L    '‹‹QC','FX››'
  DC.B    '01'                        ; Level
  DC.W    0
  DC.W    6
  DC.B    '1.01  '                    ; Version Number
  DS.W    0

; Codes
  DC.B    4,0
  DC.W    CDEA0-*                     ; -› Item
  DC.W    0                           ; -› pre-processing
  DC.W    0                           ; -› post-processing
  DC.W    D_CDEA0-*                   ; -› Description
  DC.W    A_CDEA0-*                   ; -› Attribute
  DC.W    -1                          ; end of item list

; Descriptions
D_CDEA0
  DC.W    22
  DC.B    'Window or Error return'
  DS.W    0

; Attributes
A_CDEA0
  DC.B    0,0
  DC.W    16
  DC.B    'Window          '
  DS.W    0
  DC.B    1,0
  DC.W    16
  DC.B    'Error return    '
  DS.W    0
  DC.W    -1

; Items
CDEA0
  DC.B    0
  DS.W    0
  END

; The macro HED1 sets up ‹message› at label ‹name›

; HED1 on win1_lib
;
         nolist
;
HED1     MACRO    message,name
\2       DC.W     LE\@-L\@         length of message
L\@      DC.B     \1               message
LE\@     DS.B     0                may be at an odd byte
         DS.W     0                go to a word boundary
         ENDM
;
         list
;
```

Programming in Assembler - Part 21
The Pointer Record

by Norman Dunbar

Before we dive into the PTR_GEN code again, I have a correction or two from the previous article as pointed out by my ever vigilant reader, George Gwilt. I have missed setting the timeout for the call to sd_wdef so to correct matters, change this code:

```
        moveq   #sd_wdef,d0   ; Redefine open window
        moveq   #2,d1         ; Border colour is red
        moveq   #1,d2         ; Border width is one pixel
        lea     conDef,a1     ; Console definition block
        trap    #3            ; Do it
```

To the following code by inserting a single line that sets the timeout:

```
        moveq   #sd_wdef,d0   ; Redefine open window
        moveq   #2,d1         ; Border colour is red
        moveq   #1,d2         ; Border width is one pixel
        moveq   #timeout,d3   ; Infinite timeout (THIS LINE ADDED)
        lea     conDef,a1     ; Console definition block
        trap    #3            ; Do it
```

In addition, George has made a couple of other suggestions for improving the code as a whole. You can read these elsewhere in the magazine but in summary:

* All the calls to the TRAP #3 and checking the error return can be extracted to a small subroutine and called as required.

* The timeout value in D3 is actually preserved through all the TRAP #3 calls and so need not be implicitly set after the call to SD_WDEF.

Both of these improvements have been incorporated into this article's code.
In addition to what George spotted, I have one of my own to add. The code at Exit reads as follows :

```
exit    move.l  d0,d3         ; Save any error codes
        moveq   #5,d0         ; MT_FRJOB also gives compile errors!
        moveq   #timeout,d1   ; The job to kill is this one
        trap    #1            ; Kill me
        bra.s   exit          ; We should never get here, but just in case
```

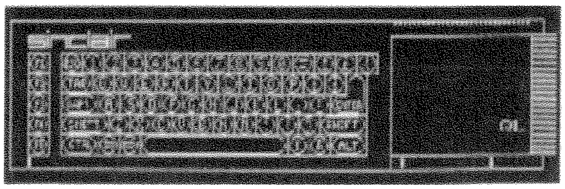
This is slightly incorrect as the line that moves the timeout value into D1 should read:

```
        moveq   #me,d1        ; The job to kill is this one
```

The reason it works is simple, the equates for timeout and me are both -1, so on this occasion, I got away with it!

Having got the errors out of the way, let us progress.

I mentioned in the previous (short) article on the Pointer Environment that the pointer record needs a bit of discussion and to this end, I've written a small pointer record diagnostic program that allows you to see what happens when you press a key and so on in a call to IOP_RPTR. The code will be shown later in this article. Note however, that it doesn't include any sub-windows yet – those are a feature of a later article.
When you make a call to IOP_RPTR you have to have A1 pointing at a 24 byte buffer, aligned on an even address, where the call will write information about things that happened, and where, during the call.



The pointer record looks like this:

Offset	Size	Description
$00	.L	Channel ID
$04	.W	Sub window number (-1 = main window)
$06	.W	X coordinate
$08	.W	Y coordinate
$0A	.B	Keystroke or button (0 = none)
$0B	.B	Key down or button down (0 = none)
$0C	.L	Event vector LSB only used
$10	.W	Window or sub-window width
$12	.W	Window or sub-window height
$14	.W	Window or sub-window X co-ordinate
$16	.W	Window or sub-window Y co-ordinate

Now, remembering back to the termination vector in the last article, you will remember that this tells IOP_RPTR when to return, so the data in the pointer record depends to a certain extent on what you set in the termination vector. In our first pointer environment example, we simply set bit 0 so we would return from the call to IOP_RPTR when a button on the mouse was pressed or a key on the keyboard (where else) was pressed.

What are all those fields in the pointer record used for?

The channel id is simply the channel ID of the window enclosing the pointer. This will not be a sub-window because sub-windows don't have an Id, they are 'simply' sections of the main window. There will be more of sub-windows in a future article.
If the window is indeed adorned with sub-windows, the second field holds a word sized sub-window number. This can be used to index into the sub-window table to fetch back the dimensions and so on of the sub-window in question. If this value is $FFFF (minus 1) then the pointer was not in any sub-windows but in the main window.
The X and Y co-ordinates are those of the pointer position within either the main window or the sub-window. The values are in pixels and both are word sized values.
The next two fields denote which key or mouse button was pressed (and released) or is being held down. For most values this corresponds to the ASCII value of the character code
- so the ESC key would be $1b or 27 (decimal) however, certain keys have different values
- a HIT with the space bar gives a code of $01 and a DO with ENTER gives $02 for example.
You will see this as we experiment later with our code for this article. A zero in these fields says that no key or mouse button was pressed/held.

Next we have the event vector which is a long word in size. Only the lowest byte is used (at offset $0F). This appears to be a bitmap of certain operations that have taken place, one or more may have caused the termination of the IOP_RPTR call.
Ok, the documentation says that only the lowest byte is used, but the documentation is very old. Things have moved on and it is possible for jobs to be sent an event, rather than generating one themselves, so it is possible that you will see data in bytes other than the lowest one.
Finally, we have 4 words defining the width, height, x and y positions of the window or sub-window in which the pointer event took place. You do not - some might say unfortunately - get the border colour and width or paper and ink colours from the pointer record.

So, now you have details of what the PE documentation has to say about the pointer record, what else can we find out about it ourselves? To answer this question and to see exactly what is stored in it after a call to IOP_RPTR, I have written the following almost useful utility to allow us to view the contents of the pointer record after an event has occurred.
I have deliberately kept it simple - as I don't want to clutter up the code with unnecessary adornments - this is not a Windows program after all! :-)

You may notice that it is very similar to our very first introduction to PTR_GEN programming as per the last article.

As ever, we start with a number of equates. None of these need any explanation, so I won't! You can experiment with the value of TermVec as described in the previous article - if you wish.

```
Me          equ     -1              ; Current job id
Timeout     equ     -1              ; Infinite timeout
OpenOld     equ     0               ; Open existing exclusive device
iop_pinf    equ     $70             ; Get PE information
iop_outl    equ     $7a             ; Outline a primary Window
iop_rptr    equ     $71             ; Read the pointer
TermVec     equ     $01             ; When to stop reading
KeyStroke   equ     $0a             ; Keystroke or button
ESC         equ     $1b             ; ESC key code
Space       equ     ' '             ; One space
LineFeed    equ     $0a             ; Linefeed
```

The usual standard QDOSMSQ job header needs no introduction by now either.

```
            bra.s   start
            dc.l    0
            dc.w    $4afb
JobName     dc.w    JobName_x-JobName-2
            dc.b    'PTR_RECORD Test v1'
JobName_x   equ     *
```

A few channel definitions and useful tables and such like come next. We are using a bigger window than the previous article as we have a bit of text to print in our window this time. The previous utility didn't do much at all, simply closing down when you clicked a button or pressed a key. This one loops around until you explicitly quit by pressing ESC.

```
ConChan     dc.w    4               ; Console channel name
            dc.b    'con_'

ConDef      dc.w    412,156,50,30   ; Primary Window width, height, x, y

HexBuff     ds.w    1               ; 2 Bytes storage for hex conversion

SpaceTab    dc.w    20,18,16,14,13,12,8,6,4,2

PtrRec      ds.w    12              ; 24 byte Pointer Record for IOP_RPTR
```

Next up we have the start of the code proper. Like last time, much of this could be considered boiler plate in that it never varies much. Obviously, my error trapping is quite simple, in the event of an error, bale out of the program. This is suitable for a small test program but in real life would need to be slightly more robust.

We start off by opening a channel to a console device. This will default the colours and so forth to a black paper and white text.

```
Start       moveq   #io_open,d0     ; Open a file or channel.
            moveq   #me,d1          ; Open for me
            moveq   #OpenOld,d3     ; Old exclusive device
            lea     ConChan,a0      ; Channel definition
            trap    #2              ; Do it
            tst.l   d0              ; OK
            bne     Exit            ; Nope, bale out.
```

Assuming the console has opened ok, we now redefine the size we want it to be and give it a red border one pixel wide. Once that has been done, we call CLS on the window.

```
            moveq   #sd_wdef,d0         ; Redefine window
            moveq   #2,d1               ; Red border
            moveq   #1,d2               ; One pixel wide
            moveq   #timeout,d3         ; Infinite timeout
            lea     ConDef,a1           ; Definition block
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.

            moveq   #sd_clear,d0        ; cls
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.
```

From this point onwards, both A0 and D3 are preserved by all the calls to TRAPs etc that we make in the program. You will not see these being set again.
Next, we have to find out if the user has loaded the Pointer Environment or not. If they have, we can continue with the remainder of the program, otherwise we simply bale out. A real program would display a message to the user telling them what the problem is and not simply 'vanish'.

```
FindPE      moveq   #iop_pinf,d0        ; Get PE information
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.
```

The PE exists and is usable. We now have to outline our primary window. This defines the area in which all pointer operations take place for this application. We also add a 4 by 4 shadow to our display to give the appearance that our application's window is floating above the screen.

```
GotPE       moveq   #iop_outl,d0        ; Outline primary window
            move.l  #$00040004,d1       ; Shadow 4 by 4
            moveq   #0,d2               ; Ignore window contents
            lea     ConDef,a1           ; Outln size
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.
```

All the preparatory work for the PE has been done, we now display a message telling the user to "press ESC to quit". As we cleared the screen earlier on, this will appear at the top of our window. We also print a string containing headers to explain what each field of the (soon to be) printed output relates to.

```
            lea     SignOn,a1           ; Message, ESC to quit
            move.w  ut_mtext,a2         ; Print message vector
            jsr     (a2)                ; Do it
            bne     Exit                ; Bale out on error

            lea     Title,a1            ; Headings
            move.w  ut_mtext,a2         ; Print message vector
            jsr     (a2)                ; Do it
            bne     Exit                ; Bale out on error
```

The main pointer loop begins here. As mentioned in the text, we are using the same termination vector as last time, return from IOP_RPTR when the user clicks a mouse button or presses a key.

```
Pointer     moveq   #iop_rptr,d0        ; Read pointer
            moveq   #0,d1               ; Initial x,y for pointer
            moveq   #TermVec,d2         ; Return on button or keypress
            lea     PtrRec,a1           ; Pointer record storage
            trap    #3                  ; Do it
            moveq   #0,d0               ; Ignore errors (We know the channel is open!)
```

When we get to this point, the call to IOP_RPTR has returned and as part of that call, the pointer record has been filled in with data. This is where we start to print it all out.
There are 24 bytes in the pointer record, so we start by initialising our byte counter to 23 – as DBRA requires. A2.L is set to the address of the pointer record and then we start a loop to convert each byte of the pointer record to hexadecimal and print it out.

```
PrintOut    moveq   #23,d7              ; 24 bytes to print out
            lea     PtrRec,a2           ; Location of bytes to print = pointer record

PLoop       move.b  (a2)+,d6            ; Fetch a byte from pointer record
            bsr.s   HexIt               ; Convert to hex in buffer at (A3)
            suba.l  #2,a3               ; Adjust buffer pointer
            exg     a1,a3               ; Buffer now in A1
            moveq   #io_sstrg,d0        ; Send bytes to channel
            moveq   #2,d2               ; Two bytes only
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.
            exg     a1,a3               ; Swap buffers back again
```

As we move through the buffer, D7 is used to keep track of how many bytes are still to be printed (minus one of course) so, at certain points along the way, we check if D7 is equal to one of the entries in our 'space table' and if so, we print a space. This is a quick and simple manner of splitting up the long string of characters that would result from converting 24 bytes to hexadecimal and printing them out.

```
SpaceReqd   lea     SpaceTab,a3         ; Table of D7 values where a space is needed
            moveq   9,d5                ; 10 values in table

SpaceNext   cmp.w   d7,(a3)+            ; Counter in table?
            dbeq    d5,SpaceNext        ; Scan until found, or not
            bne.s   LoopEnd             ; It was not found
            bsr.s   DoSpace             ; Print a single space
```

At the end of the main loop, when all 24 bytes have been converted and printed out, we throw a new line and get ready to see if we should quit or not.

```
LoopEnd     dbra    d7,PLoop            ; Do some more bytes

            bsr.s   DoLinefeed          ; Print a linefeed now
```

At this point we now start to use the data in the pointer record in 'anger'. We have printed the contents to the screen – so we will see what is in the buffer, however, if the key we pressed was ESC, we terminate the program. If it was some other key, we skip back to the start of the pointer loop and start off by reading the pointer again.
The ESC key has keycode 27 decimal or $1B hexadecimal and we look in the pointer record for that value as the key that was pressed. Remember, our termination vector said to exit when a key was pressed or button clicked so we are looking for a keystroke. It could be that we will find data elsewhere in the pointer record about our 'event' – time will tell.

```
Escape      lea     PtrRec,a2           ; Pointer record again
            cmpi.b  #esc,KeyStroke(a2)  ; Got ESC key?
            bne.s   Pointer             ; Go around again
```

This is the end of the program. We arrive here when the user presses the ESC key or if any errors occur in setting up our windows and so on.

```
Exit        move.l  d0,d3               ; Error code in D3
            moveq   #mt_frjob,d0        ; Force remove a job.
            moveq   #me,d1              ; Job id of current job.
            trap    #1                  ; Kill me
            bra.s   Exit                ; We never get here, but ...
```

This is one of George's suggested improvements, replace all those TRAP #3 calls, and error checks with a single subroutine to do it all.

```
Trap3       trap    #3                  ; Do the trap
            tst.l   d0                  ; Did it work?
            bne.s   Oops                ; Fraid not
            rts                         ; Yes it did

Oops        addq.l  #4,a7               ; Clear the return address off the stack
            bra.s   Exit                ; Bale out
```

A sub-routine to take the byte value in D6 and convert it to a pair of Hexadecimal digits in the buffer pointed to by A3. This code trashes A3 and D6 but everything else is unaffected.

```
HexIt       lea     HexBuff,a3          ; Buffer for output
            move.b  d6,-(a7)            ; Save hex byte
            lsr.b   #4,d6               ; Keep high nibble in low nibble
            bsr.s   Nibble              ; Convert nibble to hex
            move.b  (a7)+,d6            ; Restore hex byte

Nibble      andi.b  #$0f,d6             ; Keep lower nibble
            cmpi.b  #10,d6              ; Check for a-f
            bcs.s   Add_0               ; No, 0-9 only
            addq.b  #7,d6               ; Offset to 'A'
Add_0       add.b   #'0',d6             ; ASCII code now
            move.b  d6,(a3)+            ; And buffer it
            rts                         ; Done
```

A sub-routine to print out a space to the channel in A0.L. This is used between fields of the pointer record to break up the monotony of 48 hexadecimal characters in a long string across the screen. This code trashes registers as per IO_SBYTE which is what it calls to do the work. There is another subroutine here as well that prints a linefeed at the end of each decoding of the pointer record.

```
DoSpace     moveq   #Space,d1           ; Print a space
            bra.s   DoIt                ; Skip next bit

DoLinefeed  moveq   #LineFeed,d1        ; Print a linefeed

DoIt        moveq   #io_sbyte,d0        ; Send one byte to channel
            bsr.s   Trap3               ; Do trap #3 and return here if all ok.
            rts
```

And finally in this file, the two messages we print at the start of the program. One telling the user how to quit and the other is used as the headings for the columns of data produced when we run the program. Take note that there are two spaces after 'Channel' and one space before 'wide' in the following. 'KS' simply refers to Key Stroke and 'KD' is Key Down.

```
SignOn      dc.w    signon_x-signon-2
            dc.b    'Press ESC to quit...',10,10
SignOn_x    equ     *

Title       dc.w    Title_x-Title-2
            dc.b    'Channel  SubW PtrX PtrY KS KD EventVec'
            dc.b    ' Wide High Xorg Yorg',Linefeed
Title_x     equ     *
```

The way the QDOSMSQ is written and the above program takes advantage of the fact, is that A0.L is never corrupted by any of the channel handling routines. I never have to – at least in the above simple code – preserve it anywhere. It simply remains unaffected from the time the channel is opened until the job is killed off. As George pointed out in his comments on my previous article, the timeout in D3 is also preserved. The above code takes that into consideration as well.

Running the program is simple, simply EX or EXEC it and a window will appear centralised on your screen. It will be showing a prompt that says to press ESC to quit. As written the code will return from the IOP_RPTR call when a key or button is pressed, but you can experiment with different settings in the termination vector to see what happens under different circumstances.

I've written the code to put a space between each field of the pointer record when printed out on the screen. It's not the best way of doing things but is a lot easier to read than a string of 48 hex digits on screen in one line! Feel free to modify the code to print things in a better fashion if you wish!
When the code is run, move the pointer around, press various keys - try pressing keys together and see what results appear in the output. The channel Id should remain constant as should the width and placement of the window, but some of the other fields will change as you press different keys or click mouse buttons - try some together and see what you get.

As I experimented with my version of the utility, I discovered the following.
Using a termination vector of $01 - exit when a button or key is pressed:
A HIT with the button space bar sets both KeyStroke and KeyDown to $01.
A DO with the button or ENTER sets both to $02.
A normal keypress only sets KeyStroke to the ASCII code of the key. KeyDown is zero.

The event vector takes on different values according to what has been happening in the window:

After the start of the program, the pointer remains inside the hit area, a click with the mouse buttons sets the vector to $2B. This is the value when SPACE or ENTER are pressed.

If the pointer remains inside the windows as above, any other keypress sets it to $2D.
If the pointer has been outside of the window and comes back in - which it has to for the program to register events, SPACE, ENTER, HIT or DO buttons set it once to $3B. Other keypresses set it once to $3D.
If the job is 'picked' the KeyStroke is set to $08 and the event vector is set to $3D.
If the pointer is on the border then that counts as being inside the hit area for the primary window, however, if it is on the shadow, that counts as outside the primary window. So the hit area is exactly the size you defined in the call to IOP_OUTL and the additional shadow area is just window decoration.

The event vector is a single long word which records all the events which have occurred in the call to IOP_RPTR. The documentation says the following is the structure of the event vector:

Pointer Level	Bit 0	Keyclick detected
	Bit 1	Key down
	Bit 2	Key up
	Bit 3	Pointer moved
	Bit 4	Pointer moved out of the window
	Bit 5	Pointer was in the window
Sub-window	Bit 8	Sub-window split
	Bit 9	Sub-window join
	Bit 10	Sub-window pan
	Bit 11	Sub-window scroll
Window	Bit 16	Do
	Bit 17	Cancel
	Bit 18	Help
	Bit 19	Move
	Bit 20	Resize
	Bit 21	Sleep
	Bit 22	Wake

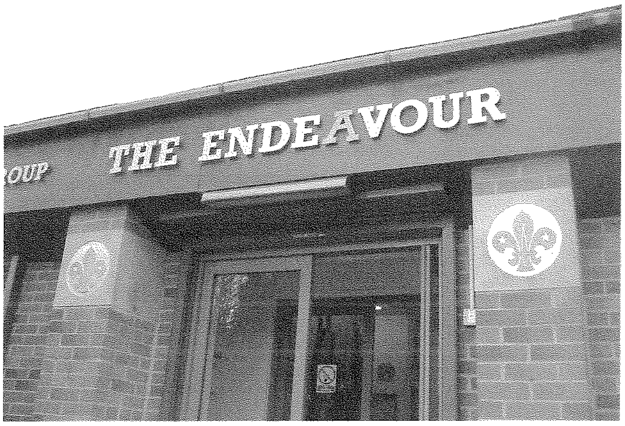
Not mentioned are events that can be sent to your job by another job. I do not have any documentation about the bits for that level and what they define. I'm sure one or two of my eagle eyed readers will let me know!
You can use the values returned from the code above to check the bits that are set in the event vector and see exactly what events were recorded while the call to IOP_RPTR was taking place.

Next time, we move on from PTR_GEN and into WMAN - at least, that's the plan.

QUANTA in the Balance

by Geoff Wicks

Quanta's Manchester shows are held in the Endeavour Scout Hut and this year we discovered how the hut got its name. Only the most hardy mortals will brave the hazards of the potholed approach road in stormy weather.

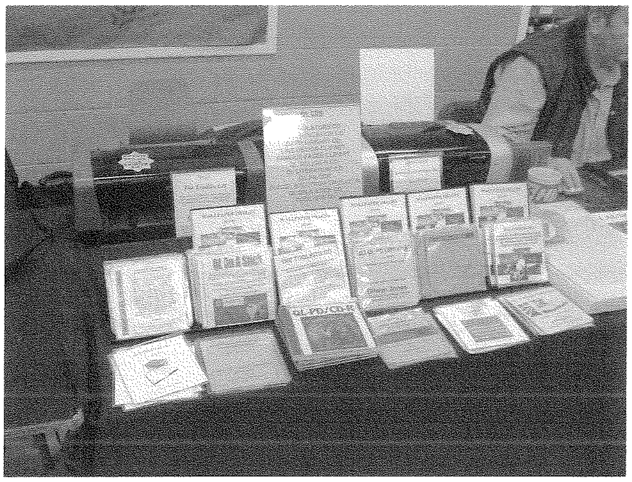


Attendance was slightly down this year, and we can only hope no one disappeared into a bottomless pit while trying to get to the hut.

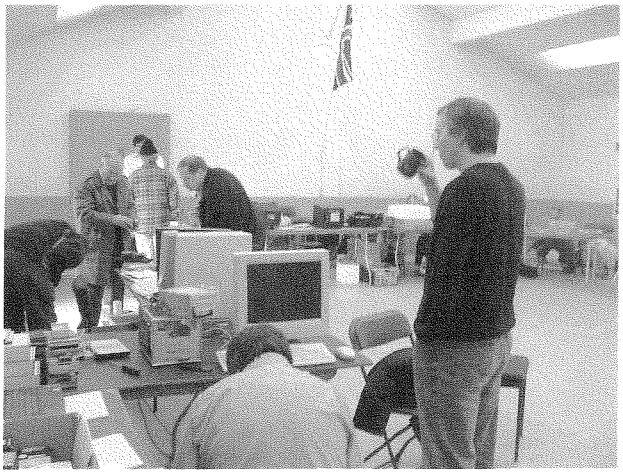
Your intrepid reporter braved not only the hazards but also went undercover to bring you a couple of images Quanta would rather you did not see.

Manchester differs from other QL shows because the traders do not like a trip to the north coupled with an overnight stay - the last full trader attendance was in 2002. NEMQLUG, who run the show, have mastered the art of running near traderless shows and, to their credit, this year they achieved the best attendance at a Quanta AGM for some time.

One trader who did attend was Rich Mellor, who has a special place in the affections of Quanta because he trades in second hand hardware on their behalf. About a quarter of Quanta's income comes from Rich's activities. Also present were the two "semi-traders" Dilwyn Jones and Just Words!



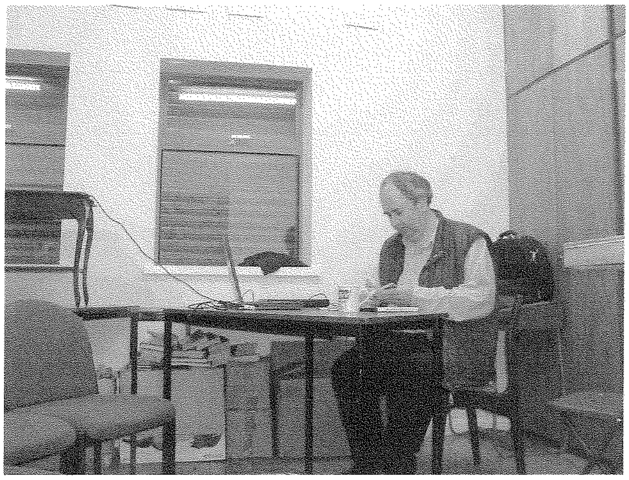
Attendance on the Saturday was in the mid twenties. Most visitors came from the Midlands and North, although Quanta members and committee from Edinburgh, North and South Wales, Dorset, Sussex and the Isle of Wight were also present.



Scotland was represented by George Gwilt who gave a presentation of his work on the pointer environment. George specialises in being awkward. Tell him something is impossible and he will soon prove you wrong, which was more or less the unofficial theme to his talk. Everyone said it was impossible to compile PE programs with Turbo before George came along. He is currently updating his TurboPTR suite to make it more user friendly than EasyPtr and adding Assembler and C compatibility. At the moment the programs are not available on the SQLUG website, but if you are interested you can contact George directly.

While at the show George gave me part 2 of the ticker article that appears in this issue of QL Today. I showed him something of the first stage of the editorial process. Assembler articles are some of the most difficult for the QL team as reformatting has to be done manually. I tried to explain to George that automatic reformatting of the text, could corrupt the formatting of the code. I soon found myself under a barrage of questions about the reasons for this and a demand that I looked at the detailed coding of the text file. We discovered most of the problems were caused by the PC having proportional fonts and the QL not. Did I really hear some mutterings from George that the QL could also have proportional fonts if the SMSQ-E licensing was minder stringent?

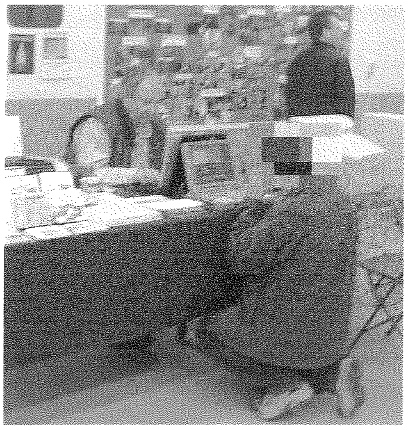
File transfers still cause headaches for QL-ers and the subject came up on several occasions during the Manchester weekend. Many QL-ers have problems transferring spreadsheets and data bases. Another problem is the physical transfer of large files from QL hard disks to PC hard disks.



The second presentation of the show was given by Dilwyn Jones. He repeated his talk on the "QL on a Stick" for the benefit of those people who were not at Birmingham. Dilwyn informs me interest in the product has been higher than he expected possibly as a result of last year's QL Today articles.

At this point I have to bring you distressing news about Dilwyn. Our more sensitive readers may like to skip the next 6 paragraphs. To my great shame I am partly to blame for this situation.

Faithful readers of my reports from QL shows will remember that two years ago I gave Dilwyn the title of "Custodian Pursuivant Extraordinary of the QL Archive". This was intended as a light hearted reference to his efforts to stop his wife from throwing away all his computer junk.



Unfortunately Dilwyn appears to have taken this title seriously and his recent elevation to the Quanta committee has added to the megalomania. As our hidden camera shows Dilwyn now expects ordinary members of Quanta to genuflect before addressing him.

Our secret camera was also focussed on Quanta's top brass. The two most senior committee members were captured by its lens examining a publication with all the enthusiasm of schoolboys examining a magazine from the top shelf of the newsagent. If you examine our photograph carefully you can see the publication which so stimulated them.

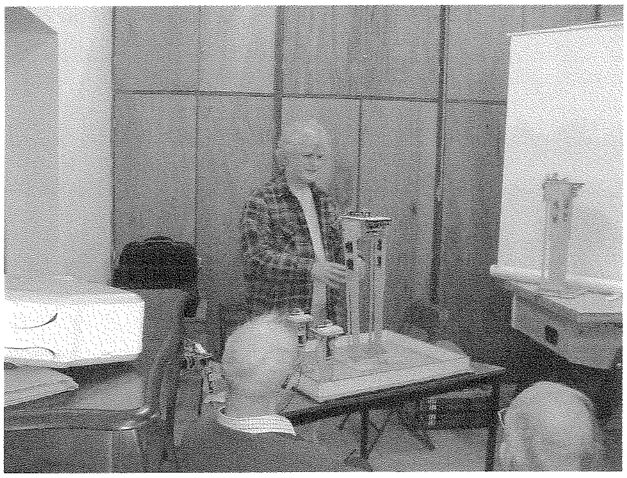
When challenged by your reporter to account for their behaviour, one brazenly replied:

"We were just looking at how you lay things out." A likely story.

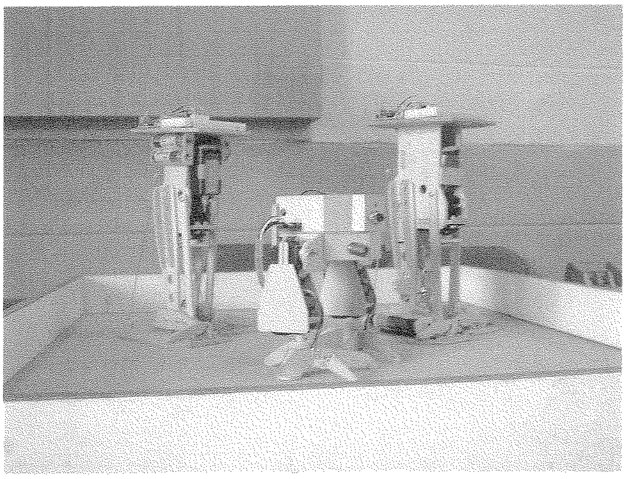
Your reporter was more interested in the layout of the table at the Pond Quay restaurant where Manchester traditionally holds its show dinners. There was an impressive display of the dark blue napkins, which are known to be an essential collectors item for the discerning connoisseur. 21 people attended the show dinner which was a multi-course Chinese banquet, including that well known favourite "Vegetarian Duck".

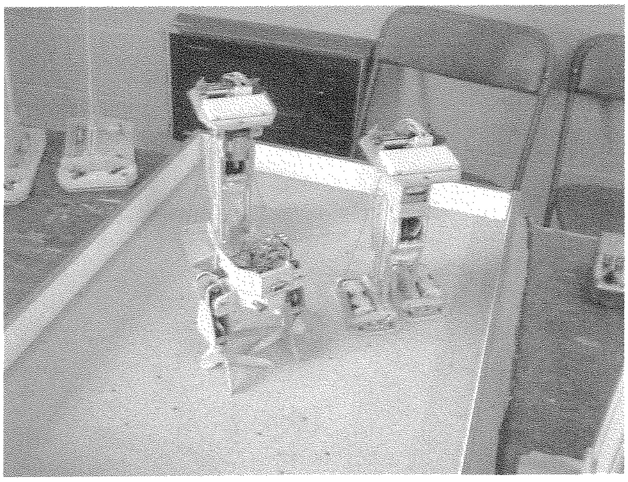
The Chinese banquet lasts for about three hours and just at the point most people think the meal is over, the main course arrives. This meal includes soup, spring rolls, duck pancakes, and prawn, chicken and beef dishes served on a rotating glass plate. The meal was a pleasant, but largely uneventful, evening in the sense that this year there were no broken glasses or disappearing table linen.

Sunday morning was slightly off topic, being devoted to David Buckley's robots. These are a feature of Manchester shows and attracted much attention especially when they were allowed to roam around the floor of the scout hut. One took an unhealthy interest in me, showing dog-like behaviour of creeping up behind to attack from the rear. I feared that at any moment he (she?) would pee on my leg.



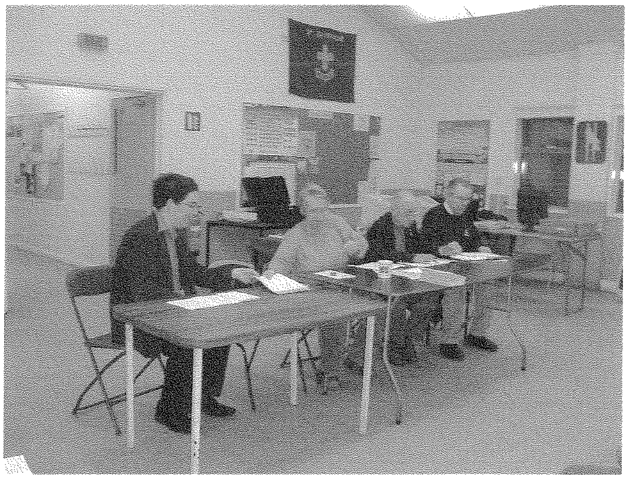
David gave an impromptu and much appreciated presentation. One of the most difficult parts of building a robot is getting it to walk. David had discovered a technique of getting the robot to lean sideways on one leg and then move the other leg forwards. This is the way most English robots walk but the Japanese use a different system based on a mathematical formula. Japanese robots have a technique of first bending their knees before moving off. David's robots have sensors which warn them if they are about to crash into something and enable them to take evasive action. One of them is able to pick himself up if he falls down.





David described how often he discovered new techniques of controlling robots by just playing with them manually and trying out new ideas. This often leads to "why didn't I think of that before" moments.

Although I described this presentation as being slightly off topic it was still relevant to QL-ing through a process Keith Mitchell describes as "sideways thinking". When Keith arrived at the show on the Saturday he demanded a table near the door because he needed an external GPS aerial. "Sideways thinking" about Hugh Room's QL Today articles had stimulated him to use GPS to obtain accurate timings for a project he was working on. Apparently there was something in David Buckley's presentation that had stimulated Keith to do more sideways thinking.



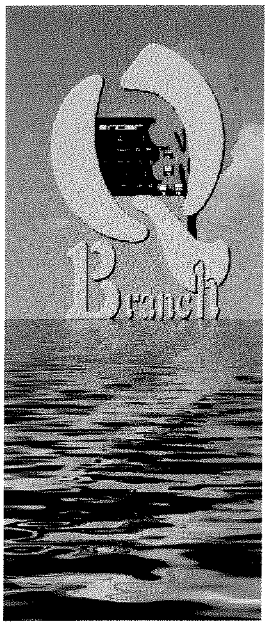
The final act of the Manchester weekend was the AGM. 20 people were present, which was a much higher attendance than in the two preceding years. The number of proxy votes, 12, was slightly down.

The meeting can probably best be described as short and sweet. The chairman was able to give a positive report of 2007 with successful shows held in Hove and Solihull; standards being maintained in the magazine and electronic publication proving worthwhile; a renewed website; and new committee members. (However, as you will have read in our news section Quanta still has a number of hurdles to surmount in the next couple of years.)

The treasurer was also able to give a positive report with the organisation breaking even. The deficit for the whole year was just £66 and the fall in Quanta's capital the same amount. Subscription income was down from £2,986 to £2,656 and magazine printing costs had risen from £800 to £1,112 because of a bill for 2006 arriving late. These were compensated by a reduction in magazine postage costs from £485 to £362 and an increase in donations from £272 to £1,167. The bulk of the donations came from Rich Mellor's sale of hardware on behalf of Quanta. Workshop costs were higher because of the additional expense of the venue at Solihull, but committee costs were almost halved.

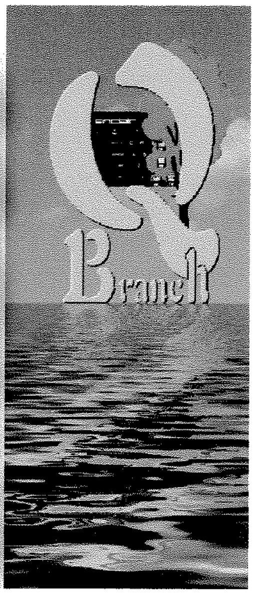
John Gilpin was also able to give a positive report as membership secretary with membership falling from 196 to just 191. However as QL Today has reported before, there is a discrepancy between different ways of determining membership trends. The official statistics show a fall of 2.6% but subscription income indicates a fall of 11.1%. QL Today suspects there was some under recording of membership in 2006, but even so there would appear to be a falling off in the rate of membership decline. Although not all the subscriptions are in for this year, early indications are that membership is holding up well.

No elections were necessary for the committee, which remains undermanned. All the old committee were prepared to continue in office and they were joined by Dilwyn Jones. Derek Stewart had been nominated for a committee post, but had had to withdraw because of a job change. (After the AGM the new committee held their first



Big QL Today Back Issue Sale

We have copies of most issues of
QL Today
£1 per issue / £7 for 10 copies
+ carriage
Fill in the gaps while stocks last!



Programs

Utilities

Fifi2	File Finder	£21.00
QSup	Utilies	£30.00
QSpread	Spreadsheet	£51.00
Cueshell 2	File Manager	£15.00
QPAC 2	File Manager & Utilities Package	£42.00
QPAC 1	Calendar, Clock, Calculator, Sysmon	£22.00
QLoad/Qref	Fast load for QDOS	£15.00
QTYP2	Spell Checker	£31.00
QLQ	Printer Utility	£30.00
SuQcess	Database	£28.00
Q-Route	Route Finder	£25.00
Knight Safe3	Backup Program	£35.00
Fractal Collection	Fractals	£35.00
QCount	Accounting	£25.00

Programming

QD 2003	Text Editor & More	£ 49.00
QBASIC	QLiberater to QD Link	£ 15.00
QLiberator	Basic Compiler	£ 50.00
QD + QBASIC		£ 63.00
QD + QBASIC + QLiberator		£104.00
QPTR	Pointer Toolkit	£ 32.00
MasterSpy	Fast Text Editor	£ 30.00
QMake	Assembler Tools	£ 18.00
QMon/Jmon	Monitor - Upgrade only	£ 22.00
BASIC Linker	Basic Library Linker	£ 22.00
Disa 3	Dissassembler	£ 34.00
QMenu	Menu Extensions & tutorial	£ 16.00
Easyptr v4	Toolkits & Programming Extns	£ 41.50
Easyptr v4	Part 3 C extensions	£ 14.00

HARDWARE

We have a rotating stock of both new and second user hardware. It is best to call or email us for details of what is available.

Recycled Items
(when available)

Super Gold Card	£email
Gold Card	£email
Aurora	£ 60.00
QXL	£ 35.00
superHermes	£ 65.00
DiRen Keyboard Interface	£ 15.00
Qplane	£ 5.00

New Items

Aurora	£70.00
Aurora w/8301 & Minerva	£80.00
8mb Rom Disq	£98.00
4mb Rom Disq	£65.00
2mb RomDisq	£39.00
mPlane	£34.00
MCplate	£ 6.50
Various braQuets	£ 8.00
Gold /Super Gold Card Batteries	£10,00

We also have a collection of standard QLs, QL Power supplies and some QL books.

Cables for the Aurora, Qubide and Super Gold Card ROMs and other QL accessories are also available from us.

Call for details

meeting at which a second new member, Tony Hill, was co-opted onto the committee.)

The meeting had to consider two amendments to the constitution. The first of these was to raise the deposit for calling a special general meeting from £300 to £500. This was in response to feeling from the floor at the 2007 AGM. This was carried with just one dissenting vote. (The dissenter was your reporter who finds it a fair question to ask why Quanta still remains so deeply traumatised by events that happened 16 years ago.)

The second constitutional amendment was sobering as it set out the procedure for winding up Quanta in more detail than previously. In spite of the recent renewed interest in Quanta, its winding up in 2009 or shortly after still remains a possibility. John Mason must stand down next year, followed by John Gilpin in 2010 and Sarah Gilpin in 2011. If no one is prepared to replace them, the organisation will have to be wound up. So far extensive lobbying by the committee has failed to find a successor to John Mason. The new clauses make provision for an interim period without a chairman, and lay down the timetable for winding up Quanta if no chairman can be found. This constitutional amendment was also approved.

From the floor the possible danger of inquorate annual and special general meetings was raised. The quorum is 10 members. Although attendance at Manchester was twice this, the previous two AGMs had had an attendance dangerously near the quorum. In response to a question from the floor about the status of proxy votes, the chairman suggested that these should count to the quorum as the members were "present in absentia". It was suggested from the floor that this should be recorded in the minutes to minimise the possibility of legal complications over this question should it ever arise in practice.

At the Solihull show last year chairman, John Mason, told QL Today he could sense a renewed interest in Quanta, and events since then confirm this. Nevertheless as the constitutional amendments have shown the organisation is not yet out of danger. Within three years all of the present officers will have to be replaced. Although a larger committee means some projects in abeyance can now go ahead, the committee will be working in a difficult environment where a further reduction of UK trading activity is on the cards.

A member of the committee told QL Today the word "change" was frequently used in their first meeting, which immediately followed the AGM. This indicates the committee are ready for the challenges ahead, but will they get more active support from the members than they are presently getting? UK QL-ers may be more dependent on the survival of Quanta than they realise.

Making a Profit from the QL

(or what happened to the traders?)

by Rich Mellor

A lot has been said over the past few months that there is no call now for commercial software and without any new hardware projects, the QL traders only continue in business because of their love for the QL and its community.

To some extent, it is true, in that there have been no new commercial software releases for over 2 years (my own QWord was probably the last one) and certainly no new hardware has been sold for many years.

However, there is still a market out there for traders, such as myself (more commonly referred to as RWAP Software) and in fact, I have made a steadily increasing turnover for the past 5 years, with even a profit to report to the tax man.

So how do I do it?

Well, the answer is diversification.

There is still a large second hand / collectors market out there, which I tackle mainly through ebay, although many of the people on ebay have gone on to purchase other items from my own website.

It also helps that I offer goods for both the QL and the Spectrum and quite often, you come across people who have one, but a passing interest in the other, so can be tempted to get one to try.

I have sold a large amount of second hand items on behalf of Quanta, and have added £1,600 to their coffers over 3 years to help fund future projects.

Keyboard membranes remain a steady seller, and in fact it may surprise people that I average at least 2 membrane sales a day, although they are mainly ones for the Spectrum.

However, through forging links with other traders I have widened the products and services I can offer, without needing one of Tony Firshman's infamous sheds to store everything; and it also means that my customers see good service, as they can treat me as a one stop shop for many of their needs.

Sometimes, it just needs a bit of lateral thinking to ensure that the QL market remains supported both for this year and into the future.

However, it could not be achieved without the active support of the users, Quanta and fellow traders for whom I owe a vote of thanks.

Editor's note: Part of Rich's idea is based on links with other traders - but that requires the other traders to stay active as well.

Letter-Box

Simon N Goodwin writes: **Default channels**

Reading through David Denham's article in the latest 'Toady' I put a couple of ideas together which might be worth noting in a follow-up paragraph. David recalled the bug in the original Qdos ROMs (VER$ "AH" and "JM") which prevented the five-parameter version of CURSOR being used for annotating diagrams, or similar tasks, in channels other than the default. But the USE command, which was omitted from Qdos at a late stage (it appeared in the Provisional User Manual given out at the QL launch in January 1984, but never any Sinclair ROMs) yet it was subsequently implemented in my freely-available DIY Toolkit extensions, first published in Sinclair QL World five years later.

USE is in DIY Toolkit volume C - to load it,

```
LRESPR USE_CODE
```

(146 bytes) then, instead of:

```
CURSOR #chan,x,y,h,v
```

original ROM aficionados can use:

```
USE #chan:CURSOR x,y,h,v:USE
```

In fact you only need to reset the default channel to #1 (which is what the third statement, with no argument, does) when you've finished performing a sequence of operations on #chan, so the resultant code may be shorter too, if there are several other operations using #chan, marginally faster as the channel parameter need not be specified each time, and will work on all versions of Qdos.

USE works with the following standard SuperBASIC commands:

ARC, ARC_R, AT, BLOCK, BORDER, CIRCLE, CIRCLE_R, CLS, CSIZE, CURSOR, DIR, FILL, FLASH, FORMAT, INPUT, LINE, LINE_R, OVER, PAN, PAPER, POINT, POINT_R, PRINT, RECOL, SCALE, SCROLL, STRIP, UNDER, WIDTH, WINDOW

It also works with these Turbo Toolkit extensions:

LIST_TASKS, CURSOR_ON, CURSOR_OFF, EDIT$, EDIT%, EDITF, SET_FONT

and these QJump Toolkit ones:

CHAR_INC, CHAR_USE, CURSEN, CURDIS, DLIST, EXTRAS, JOBS, PRINT_USING, STAT, VIEW, WDIR, WSTAT

and these Speedscreen extensions:

_FOUNT, _XSTEP, _YSTEP

DIY Toolkit is one of many collections of SuperBASIC extensions on Dilwyn Jones's web compilation at:

http://www.dilwyn.uk6.net/tk/

It's in the archive of the first eight 'volumes' of the series:

http://www.dilwyn.uk6.net/tk/diytk1.zip

It's worth looking at Dilwyn's main page as that collates many other toolkit commands, most of which - if they take a channel parameter - work just as well with USE. The accompanying Quill document, VolC_USE_DOC, lists a few extensions that do not default to #1, and notes a few graphics commands which store state from previous operations (such as co-ordinates or the turtle pen position) with the current channel, and may seem to behave oddly with USE unless you're aware of the implied history.

Simon also writes: **Browsers**

I feel another clarification for Volume 12 issue 3 is due; Netscape was neither the 'original web browser' nor the first widely-used or ported one. After Tim Berners-Lee's original 'WorldWideWeb' application for the 68K-based NeXT computer, Erwise was developed for Unix systems at Helsinki University, followed by ViolaWWW, with advanced support for graphics, animation and applets; then Arena, written by Dave Raggett at HP in Bristol, UK. The longest-lasting of the early browsers was the pioneering text-mode hypertext browser, Lynx, written at the University of

Kansas. Version 2 of Lynx was adapted to use Tim's then-new HTML standard and still runs on many systems, including Qdos. Standards-compliant web pages should be usable with Lynx, through the required ALT tags, for example.

Only then, around 1993, did PCs and Macs start to catch on to the Web, with the development of Mosaic, the direct precursor in both code and functional limitations of both Netscape and Internet Explorer. Mosaic was a read-only browser written at the NCSA 'National Center for Super-Computer Applications' in Illinois, USA by people who went on to found Netscape. Marketed as if itself was 'the web', Mosaic won critical mass, while turning the web from a collaborative editing and sharing system to one more suited for few-to-many publishing.

If anything since Tim's NeXT prototype deserves the title of 'original web browser' I feel it's probably Mosaic. Microsoft's Internet Explorer still displays acknowledgement to NCSA, using code licensed through a commercial spin-off called Spyglass, and Netscape was based on the same source. I still have the early port 'AMosaic' on the Amiga where I'm typing this - it was the first browser I used, though it struggles with modern web pages, lacking CSS, tables and other extensions.

Mosaic caught on, though at about the same time Navipress arrived on PC and Mac, with the integrated editing capability of earlier browsers. Sadly, few modern browsers - apart from the WWW's own sadly-neglected Amaya - make it as easy to create as to read web-pages, which was a key part of the original web concept. But in a basic way Wikis, like the QLs own, reintroduce this concept, as at:

http://www.rwapadventures.com/ql_wiki

I recommend Tim Berners-Lee's book 'Weaving the Web' (published by Texere) for anyone who'd like to know how the Web came about and how the concept and commercial reality have diverged.

Getting back to the stats in Geoff's obituary for Netscape Navigator, the reason so many people visiting the Just Words! website appear to be using Netscape - 42%, apparently - is because Netscape, Firefox and several other browsers, including many configured to report that they're not Internet Explorer but which set out to be in the relatively standards-compliant mainstream, report their browser type as 'mozilla'. That's the internal name used by Netscape, then Firefox, and others spoofing that original de-facto standard for compatibility reasons (such as, optionally, my Amiga's IBrowse and AWeb). These are all reported as 'Netscape' by common net stat tools, but they could be anything other than Internet Explorer or even IE in disguise.

In fact Microsoft got the ball rolling on this one by configuring an early version of Explorer to pretend to be Netscape Navigator by returning a 'user_agent' name containing mozilla, the name of the Netscape company mascot, as part of their policy of imposing their calculatedly non-standard Explorer on Windows victims. Opera, one of the main casualties of the enforced bundling of Explorer with PCs, documented this abuse to the EU as part of their case supporting the latest billion dollar fines on Microsoft for restraint of trade.

For more on the murky world of User Agent spoofing, see:

http://en.wikipedia.org/wiki/User_agent

and for the definitive WWW and HTML standards:

http://www.w3.org

Suffice to say there are more browsers than you might think, and tools that attempt to find out which ones people are using are easily and often systematically fooled, for good or bad reasons. I believe most of the people Geoff's site reports as using Netscape are probably using Firefox. Netscape 9 was in fact just a version of Firefox re-skilled by AOL - but such statistics should be taken with a big pinch of salt.

H. Froggatt writes: No Go LINUX

I was somewhat surprised to read the small article on page 6 of QL Today, Volume 12 Issue 3 suggesting that QPC would not run under Wine on a Linux system. This note is being written in Quill (Xchange) running under QPC using Wine under Linux. The machine is fairly powerful (dual core AMD 4800). The Linux distribution is Ubuntu 8.04 (but QPC ran equally well under Ubuntu 7.10 for this machine). It is perhaps relevant that the version of wine now being used was changed to 0.9.59 when I upgraded from Ubuntu 7.10 to Ubuntu 8.04. I am using QPC II 3.34 Beta 3 (which is another story).

I have also tried this version of QPC on a less powerful machine with an AMD Duron 700 processor. Initially I had the key repetition problem mentioned in the QLT article which made QPC virtually unusable. Minor improvements in usability resulted from setting the windows version to NT in configuring Wine. Using full screen (in the QPC configuration screen) also seemed to help a bit. However, a key improvement resulted from altering the Power Management setting in the QPC configuration screen. This made a vast improvement and with careful, not too fast typing QPC

became quite usable. Too fast typing or holding down keys too long did however still result in some character repetition problems. At this stage I was trying out Mandriva 2007 (Free) and Ubuntu 7.10 Linux distributions with similar results.
Of even more interest is that I have in the last few days upgraded the Ubuntu to 8.04 on the older machine and as a result of this (or the concurrent upgrade of Wine) the QPC character repetition problem has disappeared when using Ubuntu. The keyboard response is perhaps a bit sluggish, but nothing to worry about.

Steve Poole has spotted an error in his "Off Screen Drawing" article in the last issue of QL Today. He writes: "Will readers please note that there is an error in the article 'Off-Screen Drawing' in the last QL Today. On page 52, please strike out the sentences "But to no avail. The 30 second delay remains 30 seconds," and also change 'cannot' to 'can' in the following sentence. This error arises from a typographic error in `LINE 270` where `IF sc>256: n=sc` should read `IF sca>256: n=sca`
If you then reread the article, it will be clear that it is advantageous to write code to eliminate off-screen drawing after all!" Steve would welcome some feedback on this article, particularly from any one with knowledge of the inner workings of SMSQ-E. Steve has also cast an eagle eye over Hugh Rooms' "Qtest_bas" program on page 43 of the last issue. He writes: "A comment on page 43, lines 600, 800 & 910. If you refer to Jan Jones's SuperBasic Handbook, page 39, (footnote), you will see that x==0 or i==0 can never be true....if Hugh amends them x+1==1 or i+1==1, his IF clauses will work as he expects. Dilwyn's Random Access Files article in the last issue was also interesting as it reveals fundamental principles of data bases."

Hugh Rooms replies:
"Thanks to Steve Poole for pointing out my carelessness in using "IF x==0" instead of "IF (x+1)==1" (9) several times in my last program, lines 600, 800 and 910. I should have been more diligent, although the program does work in these cases -- that's my excuse anyway."
Hugh has spotted a couple of other errors in his earlier articles:
"Unfortunately, in my last article in Volume 12 Issue 3 March-May 2008, page 39, the Internet addresses were missed out of some of the references, so I have repeated them with the same numbers, (1) to (8), at the end of my article in this issue.
On page 19 of Volume 12 Issue 1 Sept-November 2007 I say that "... the time taken for the signal to travel; divide by the speed of propagation and you have the distance ...". It should of course be "multiply" not "divide". "

George Gwilt writes: re Norman Dunbar's Article on Assembler - Part 20

[Editor's comment: Norman has replied already - and we merged his replies into George's letter - marked ND and printed in bold italics, like this text here.]

Probably the best language for writing PE programs is assembler. It is therefore very useful that Norman has embarked on this subject. This will be particularly welcome given the success of his EasyPTR Tutorial in QUANTA May to July 1994. However, I have a few comments on this first of the assembler PE series.

Trivial Comments

1. I cannot understand why Norman found compile errors (presumably using GWASL) with the constants io_open and mt_frjob but not with the constants sd_wdef and sd_clear. All these constants are in the file gwass_equ2_sym supplied in the file gwaslp04.zip on the SQLUG site. The version of GWASL in that zip file is 2.01, which is the latest version.
[ND] I have been using GWASL since the beginning of the assembler series and have never had any problems like this before. I tried everything I could to make the error messages go away - with no success. I've used these equates before and had absolutely no problems assembling the file. I'm as puzzled as George is but as I needed to have working code for the article, I decided to leave it as it was in the compilable and running state until further investigations could be made.

I have checked that all four of the constants are in gwass_equ2_sym and I have also assembled without error Norman's program using GWASL and using the names of all four constants.
[ND] I too have checked and as I said, I've used these before with no problems. This time though, I couldn't make the error messages go away.

2. When I was typing in the program I got more and more annoyed at having to repeat several times:

```
        trap      #3
        tst.l     d0
        bne.s     exit
```

What I would have done is to replace each occurrence by:

```
        bsr.s     tp3
```

and write the subroutine tp3 as:

```
tp3     trap      #3
        tst.l     d0
        bne.s     tp3_1
        rts
tp3_1   addq.l    #4,a7
        bra.s     exit
```

If there are n occurrences, the first way needs 3*n instructions: the second way needs 6+n instructions.
There is a saving if 3*n‹6+n, ie if n›3, which it is.
That is not to imply any criticism of Norman's program, which indeed will be faster though longer.
[ND] Yes indeed, George is correct (as ever) and his version would have been a better solution. I'm a wee bit puzzled as to why I didn't do it that way myself. Probably too busy concentrating on figuring out what I was doing with the PTR_GEN calls! I have incorporated George's code into the code in the current article.

3. I have always taken note of the usual preservation of register contents in traps #2 and #3. Norman has, quite rightly, assumed that A0 will continue to hold the channel ID set by io_open at "start". However, he has not taken advantage of the fact that D3 will continue to hold -1 from the first time it was set. By the way it should have been set for sd_wdef especially as it will be zero at that point, having been set zero for io_open.
Thus the timeout should be set before the first trap #3 and need not be set again thereafter.
[ND] This is correct, the timeout should have been set for the call to SD_WDEF as George notes above. To this end the code that follows:

```
moveq #sd_wdef,d0   ; Redefine open window
moveq #2,d1         ; Border colour is red
moveq #1,d2         ; Border width is one
                      pixel
lea   conDef,a1     ; Console definition
                      block
trap  #3            ; Do it
```

Should be changed to this :

```
moveq #sd_wdef,d0 ; Redefine open window
moveq #2,d1       ; Border colour is red
moveq #1,d2       ; Border width is one
                    pixel
moveq #timeout,d3 ; Infinite timeout
lea   conDef,a1   ; Console definition
                    block
trap  #3           ; Do it
```

I have to say that I never noticed that the timeout was preserved through each TRAP #3 call - but I have noticed now and the code in the current article has been modified to take this into consideration.

4. At "FindPE" Norman uses iop_pinf with the remark "Get PE Information". In fact, if ptr_gen is present, on exit D1.L will contain its version number. If wman is also present A1 will contain the window manager vector which is needed if you want to use any of the vectored routines. If wman is not present A1 will be zero.
[ND] Yes this is true and I was aware that the call to IOP_PINF will return more than just an indication of whether the PE is found or not. For the purposes of this small example though, I don't need versions or vectors. That will be coming in a future article when I delve into the WMAN interface.

A Slightly Less Trivial Comment

5. Norman has started this series by examining three of the trap #3 routines in ptr_gen. He has, as it were, opened the PE bonnet and started to examine the exposed engine parts. This can be enthralling, but it does not help someone who wants to drive the vehicle. To do this you need to know how to use the various levers - handbrake, clutch and so on. PE's levers are the vectors of wman.

The only trap #3 routine needed for relatively simple programs is iop_pinf, which is required to access wman's vectors. All the other trap #3 routines form the building blocks of the wman vectors which themselves cover all but the most esoteric needs of a programmer.

[ND] As I mentioned in my first article, I too am learning about the PE and assembly language and I'm grateful to anyone who points out things that I'm either getting wrong or could be done in a better/quicker/more efficient way. The end result is a better series of articles. Thanks George.

[And finally, Editor's comment: Wow - that was the largest letter-box section we ever had. Great to see that there is so much feedback. And also good to see for the authors, that their hard work of writing articles is appreciated by the readers by the fact, that they actually read their articles, learn from them, use them ... whatever.]

QL TODAY Toolkits Cover Disk

by Dilwyn Jones

This cover disk is a sampler of the content of the QL Toolkits CD, which contains a large number of toolkits (basic extensions) and SuperBasic/SBasic programming utilities.

The full QXL.WIN format CD (also with zipped copies of all files for users unable to use QXL.WIN CDs) is available from Dilwyn Jones or Q-Celt Computing and contains many more programs and toolkit files than we can fit on a single floppy disk like this. For a copy of the full CD, send 5 pounds (includes UK postage) to:

Dilwyn Jones, 41 Bro Emrys, Tal-y-bont, Bangor, Gwynedd, LL57 3YT
Email: **dilwyn@uk6.net**

For orders from outside the U.K. send the equivalent price in your currency plus postage to:

Darren Branagh, Q-Celt Computing, Funionsa, Funshinagh, Cross, Co. Mayo, Ireland.
Telephone: +353-9495 45033 Email: **dbranagh@kingfisherclub.com**

The CD itself is a freeware collection of QL software and may be freely copied for all QL users. Please observe any copyright notices or restrictions placed by individual authors regarding use or distribution of their work.

The following toolkits are supplied on this sampler cover disk. All are fully working versions with no restrictions.

- **Display Code 2** Display_code extensions - aids to writing programs for high resolution and high colour systems. Originally published as an article in QL Today, this is the latest version 2.02 with a number of enhancements. ***Instructions: display_doc***
- **DJ Toolkit** Norman Dunbar's DJToolkit v1.15. Can be used from BASIC or compiled programs (the toolkit is easily linked to Turbo or QLiberator compiled programs) and includes many useful extensions which work well on modern systems. Formerly a commercial product from DJC. ***Instructions: DJTK_doc***
- **Environment Variables** Environment Variables extensions v1.07. Environment variables are basically global variables used to control certain aspects of how your system operates. Using these, you can set up certain values or settings in your boot programs and your programs can check the values and settings you have set. ***Instructions: ENVIRON_DOC***
- **Exists** Extension to check if a given extension name exists on your system. ***Instructions: EXISTS_TXT***
- **HB Utils** Hotkey and button utilities from Richard Kettlewell. INSTRUCTIONS: HBUtils_txt JTool A jobs toolkit with several job control extensions. ***Instructions: JTOOL_TXT***
- **MegaToolkit** 1.13 Huge number of extensions in this comprehensive toolkit by M. A. Crowe. A formerly commercial product. ***Instructions: Manual_Doc***
- **Path** PTH (path) device driver extensions. ***Instructions: PTH_TXT***
- **QView Toolkit** QView Toolkit v1.3, a small toolkit with a number of useful extensions from Laurence Reeves and the team at QView. ***Instructions: QVTK_TXT***
- **Redir** Redirection device driver, which allows you to assign other names or aliases to devices or files, for transparent redirection of files. ***Instructions: REDIR_TXT***
- **Strings** A set of string handling extensions, offering case conversion, string reversal, and pattern matched '?' string search. ***Instructions: strings_doc***
- **Sub** SUB is a name substitution device driver from Phil Borman. ***Instructions: sub_txt***
- **Tiny Toolkit** Tiny Toolkit v1.11, more extensions than the name implies! Original toolkit by Matthias Leidig, updated by Rich Mellor. ***Instructions: TinyTool_Doc***
- **Turbo Toolkit** Version 3.38 of the well known Turbo Toolkit. Includes QDOS and SMSQ versions, as well as a version which can be linked into Turbo compiled programs. ***Instructions: turbotk_txt***
- **WD Extensions** WD extensions for wildcard Delete, Dir and Stat. ***Instructions: WD_TXT***

Asus eeePC

by Tobias Fröschle

A venture into mobile QL computing with the Asus eeePC

It has recently been mentioned both on the QL mailing list and in QL Today: The new Asus eeePC seems to be a perfect partner for mobile QL computing. I have bought one some weeks ago and would like to present my findings with regards to using this device as a platform for QL emulation here.

First, let's have a look at the features as published on paper:

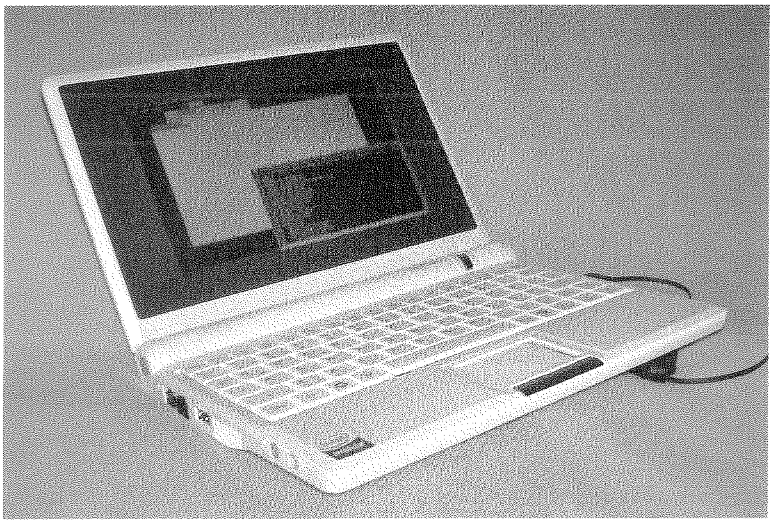
- 900MHz Intel mobile Celeron CPU running at 600MHz (Runs at reduced speed to conserve battery power)
- 512MB RAM, expandable (memory as supplied must be replaced, there is only a single SIMM slot on the board). Maximum expansion up to 2GB
- 2, 4 or 8 GB solid state Flash disk, acts as a hard disk replacement (I have yet to see samples of the 2 or 8Gig models in Europe. I would expect there are only 4Gig models in the shops)
- Sub-SVGA Display (800 x 480 pixels)
- 1 Ethernet (10B-T)
- Wireless LAN (up to 54MBits)
- 1 SDHC (SD-Card) slot
- 3 USB-2.0 ports
- Mic/Earsockets for sound
- 1 VGA socket for external monitor (supports higher resolution than the inbuilt LCD)
- Onboard Modem (not available in most European countries)
- Comes pre-installed with Xandros Linux, drivers for optional installation of Windows XP are supplied on CD

High-Street price: 299 Euros and about £200 in the UK. Unlikely to go down soon, as there currently seems to be a real run on the eeePC.

What you get

At the first few glances, I was really surprised how small this computer is. From pictures on the internet or catalogues, the eeePC may look like a normal laptop, just a bit smaller. In reality, it's a whole lot smaller, about half the size of a "normal" laptop. This has some dire consequences regarding size of keyboard and screen, as we'll see later in this review. There are some other consequences: My wife and daughters found this little computer so cute, I could hardly get my hands back on it after I had showed it to them.

The next thing you see is the small keyboard. It is actually so small, that it is barely impossible to use it to touch-type (at least my fingers just don't quite fit). So if you're a touch-typist, you'll have to revert to the good old 2-Fingers system.



What you'll also realize very quickly is that the display is small as well. If you do not have perfect eyesight, I really cannot recommend the eeePC. It just about works in Linux or Windows, but problems arise with QL emulators, especially with uQLX, that cannot scale the QL screen: The physical pixels on the eeePCs screen are so small, that a QL 6x10 character ends up like 2 mm wide and about 3 high. It's different with QPC2, where you can scale up the screen to use larger characters (albeit losing screen real estate).

What you feel if you're using the eeePC literally as a laptop is that it's getting warm. You don't burn your private parts, but you can really feel the eeePC draining its batteries. Another disturbing thing is the display lid that seems to be too heavy in relation to the keyboard part. The eeePC has an annoying tendency to tilt back onto its display if you don't hold it.

Battery life is about 2 hours using Windows XP. Probably a bit less than what you would expect from such a small PC. I am under the impression that with Xandros Linux it used to be more like three hours.

Xandros Linux

The pre-installed operating system is Xandros linux, a Debian derivative. It perfectly hides anything Linux/Unix-specific from the user, and is basically a closed system. (Even installing new applications apart from the inbuilt ones is basically impossible.) But the applications it comes with are just about enough for most things you'd want to do with such a computer. Unfortunately, there's no QL Emulator among them, though.....

eeeXUbuntu

This was the OS I installed first, because I wanted to run uQLx on the eeePC. For someone used to Linux, this is very straightforward: The eeePC can boot from a USB device, you just go ahead and install. from there. For non-Linux-experts, I really wouldn't recommend eeeXUbuntu, however. UQLx works fine on eeeXubuntu, once you manage to generate a working binary from the sources. One disadvantage of uQLx is that it cannot scale the QL screen. So you're stuck with 6x10 pixel characters, and while travelling on bumpy roads, rails or airways, that's really hard to read. And anyway: I couldn't get any of the more recent uQLx-tarfiles to actually run (This is not an eeePC specific problem, however, and I would not blame uQLx for that - It's probably all my fault). So I took the challenge and installed Windows XP.

Windows XP

It took me half a day to install Windows XP on the eeePC. Note you need a USB CDROM drive in order to install Windows XP. And don't even dare to start on a 4G model like mine if you don't have at least some external memory (1Gig SD or USB). The operating system and drivers just fit into 3Gig, the rest is needed during installation of Service Pack 2. You should reserve the internal "disk" for OS uses only and store your own data and programs somewhere else.

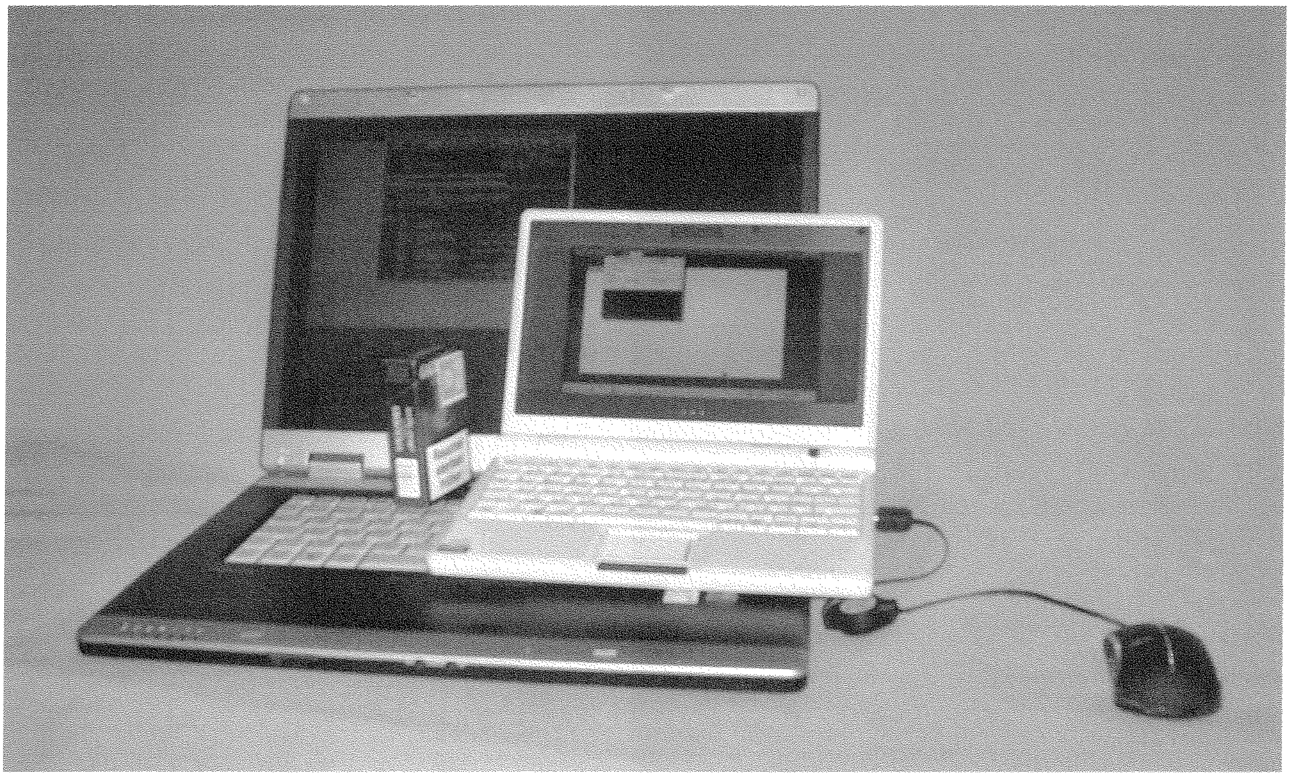
I chose an 8G SDHC card for that purpose, and that works fine, although it's a bit slower than USB storage (but has the advantage that nothing protrudes from the socket, so you can leave it installed during travel without risk of breakage)

You should make sure that you turn off the Windows XP swapfile to keep disk usage down. The only issue I have with Windows XP is that it takes the eeePC ages (like 1 minute) to restore from hibernation. For the sake of my internal flash disk, I've turned hibernation off now.

QPC

Not much to say here. It simply works. Regarding speed I can't tell the difference between my (fast) desktop PC and QPC on the eeePC. So that would mean it's fast enough.

To wrap it all up, the eeePC is a nice toy and when you run QPC2 on it, it's even nicer. Gives you the ability to take your QL stuff just about everywhere.



The Transverse Mercator Projection

by Hugh Rooms

Introduction

In this article and the next I hope to show how Mercator's projection is adapted for accurate local, e.g. national, maps. As always I have tried to be correct, but please check carefully before you rely too much on what I say here.

A cartographer looking for a projection to draw a map of a substantial portion of the Earth's curved surface on a flat sheet of paper will choose one that gives the least distortion for his (or his client's) purpose. Many different projections have been devised for special maps, for example those that have equal areas the same size on all parts of the globe, for pictures of population distributions, natural resources, and a lot of etceteras.

Navigators, gunners (ordnance), surveyors, and roamers about the countryside need a map that provides correct bearings (directions) and distances; and, as we saw in my previous article, Mercator's projection preserves directions overall, and distances over a small area.

Mercator does so by having the meridians, which converge on the poles on the Earth, spread out so they are parallel on the paper, so East-West distances are stretched out as they get further from the Equator, and, consequently, the distance between the parallels of latitude have to be increased to compensate. But, looking at a map drawn using Mercator's projection, we see that the distortions are least for a narrow band along the Equator, which is fine for countries in that region, but less so for the UK and others at high latitudes. It is pretty useless for countries like Norway and Chile.

Turning it on its side

Just to be clear, I assume throughout these articles that North is always going to be upwards (y-direction) on the map, and what we are after is an (x,y) point on the paper corresponding to the (longitude, latitude) co-ordinates of a point on the Earth.

For the UK which is twice as long N-S as it is E-W, and other countries that are elongated N-S, to get the least distortion the solution is to turn Mercator on its side (Transverse) so that a 'virtual Equator' now runs N-S down the middle, and most of the country is close to it. The distortions are now at the sides instead of the top and bottom, so they are closer to the new 'Equator' and consequently smaller.

For a normal Mercator projection we have the converging meridians up and down the map, perpendicular to the Equator and parallels of latitude, across the map, parallel to the Equator. However, with the projection on its side it's all the other way round, so we cannot apply the Mercator formula directly; but it can, of course, be done indirectly, so here goes.

The 'Aha!' moment for me was the realisation that we can invent a new co-ordinate scheme that does meet the criteria for Transverse Mercator to apply. To get my mind round it all I made a couple of models, the early versions appeared in the photos on page 28 of Volume 11 Issue 5 June-August 2007. I first drew the grid of normal latitude and longitude roughly on a 'globe' -- a small plastic ball, shown again in figure 1. We can draw another grid on the globe, a Transverse grid, of lines parallel to a chosen N-S line (the Prime Meridian), and lines at right angles to these that do converge to a point; this grid too is on the globe in figure 1.

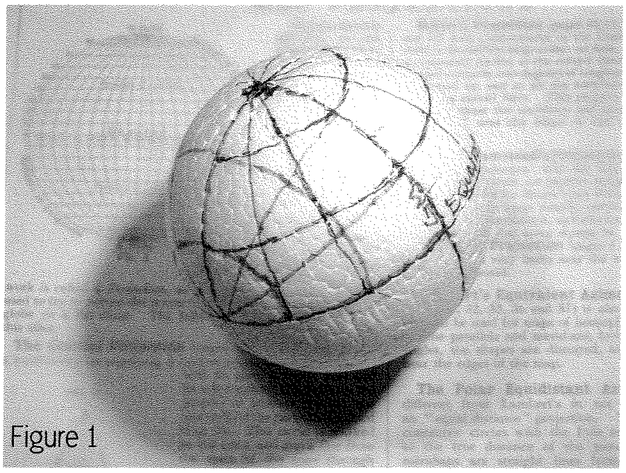


Figure 1

Here's a tidied up version of my reasoning from then on. Figure 2 shows a point P in (known) convential co-ordinates of longitude and latitude. P will also have (different) co-ordinates in the new Transverse scheme, and these will now meet the criteria for applying Mercator, but sideways.

A formalised version of the transverse grid is in figure 3. I have prefixed the equivalents of latitude etc. in this Transverse scheme with a 'T' so first we have the TEquator running N-S along the chosen Prime Meridian, which I have made to coincide with the Greenwich Meridian for convenience -- it avoids having to keep adding or subtracting an offset. Lines of constant TLatitude

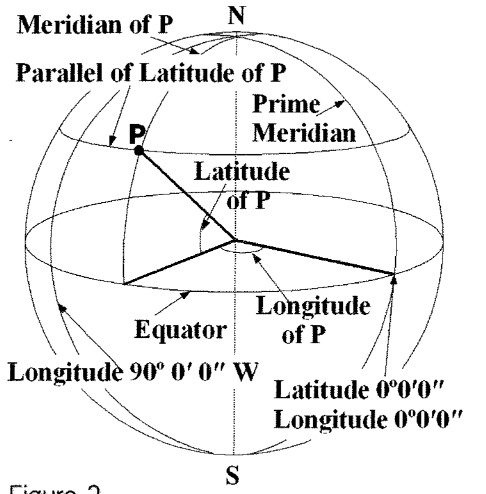


Figure 2

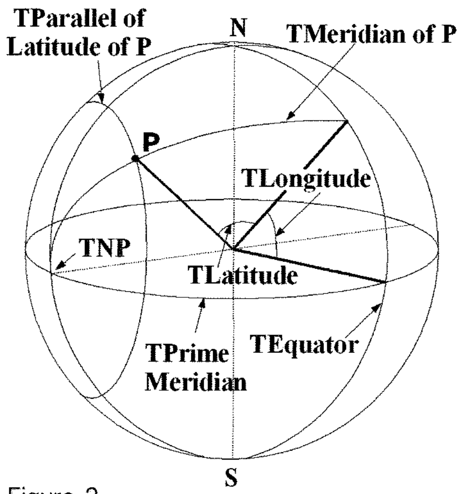


Figure 3

run parallel to this TEquator and one of them goes through P. Next, the TMeridians start off perpendicular to the TEquator, and converge on a 'North' TPole or TNP, which must be located as shown at 90 degrees W on the (real) Equator. Again, one particular TMeridian will go through P giving it a TLongitude. So these two, the TLatitude and TLongitude though P specify its position on a grid of Transverse co-ordinates.

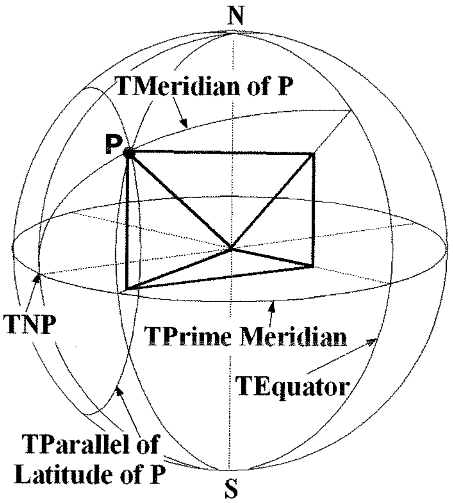


Figure 6

We also see that we will be able to apply Mercator to these Transverse co-ordinates, when we eventually get them, with the difference that the parallel lines made by pulling the TMeridians apart as they approach the TN pole now run horizontally on the paper. The TLatitudes are already parallel; but now they are drawn vertically, and their spacing increases horizontally as we get further sideways from the TEquator. The distortions in the normal Mercator projection increase N-S, and in the Transverse case E-W, or, if we think of distances, compared with the scales on the 'Equators': for normal Mercator the E-W distances get stretched as we go N or S, and for Transverse Mercator the N-S distances get stretched as we go further E or W. There had to be formulae relating the normal Latitude and Longitude to these new Transverse counterparts, and, I found them in published texts (3), but no explanation of how they were obtained. Once again I resorted to a model to get an idea of how to do it. Figure 4 is a picture of the latest version, and you can make one if you photocopy figure 5 on to card or thick paper, (I used 90gsm paper) cut it out, and assemble it. After contemplating this model for a while, I could at last see how the pairs of co-ordinates could be related.

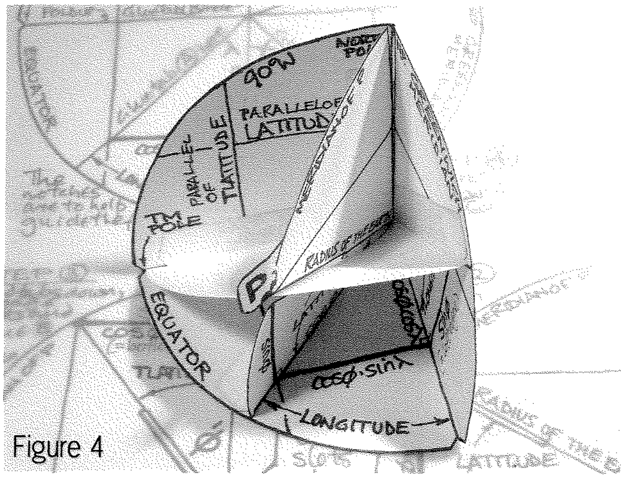


Figure 4

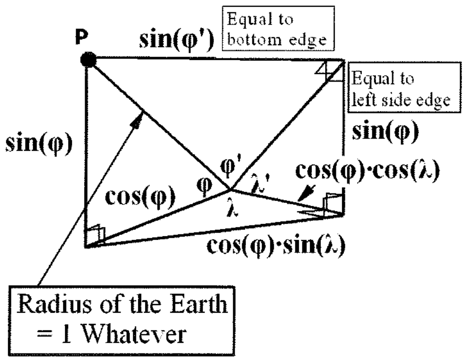
[Editor's comment: QL Today has done the copying for you; you will find figure 5 on a separate sheet, so you can start assembling it right now, if you like.]

Figure 6 and 7 formalise this, where we see that we can drop perpendiculars from P on to the old and new Equatorial planes, add two more lines at right angles on the planes themselves, and join the points to form a rectangle. Lines from the corners into the centre of the globe give us a rectangular cone; the four angles at the base of this cone are the pairs of normal and transverse Latitudes and Longitudes; and now it is fairly simple trigonometry to relate them. That's the essence of it.

To correspond with the notation where I found the formulae, and in my previous article, I call the Latitude φ, and the Longitude λ, and their Transverse counterparts the same but primed. For the formulae to translate from normal co-ordinates to transverse, we want TLatitude and TLongitude in terms of the original, known, λ and Longitude of P; that is φ' and λ' in terms of $\varphi$and λ.

Calling the lines to the centre the 'edges', only one of them is known, it is the top left edge - the radius of the Earth ending at P, and, as in my Mercator article I treat this from now on as one Whatever. The sides of the cone form four nice right angled triangles, so we can dive into some trigonometry, which I've put on a larger view of the rectangular cone in figure 7, and on the cut-out model.

The rectangular cone enlarged, with angles, sides and right-angled corners marked.



$$\sin(\varphi') = \cos(\varphi)\cdot\sin(\lambda)$$

$$\tan(\lambda') = \frac{\sin(\varphi)}{\cos(\varphi)\cdot\cos(\lambda)}$$

Figure 7

The top face of the rectangle has the Tlatitude as its inner angle, and the (one Whatever) radius as its hypotenuse, so the top front of the cone is (one Whatever)*sin(Tlatitude) or just sin(φ'), giving us a route to find the φ' we need. I will drop the Whatever (but all of the formulae for the other edges of the cone are implicitly multiplied by this value) and go straight to the Greek symbols, for the rest of the calculations.

Starting from the known top left edge, we have the bottom left edge equal to cos(φ). This is the hypotenuse of the bottom face, with the Longitude λ as its inner angle, which makes the front bottom cos(φ)*sin(λ); and the top front edge is the same. Combining these with sin(φ') in the top face, we now have the Tlatitude, φ', in terms of the given Latitude and Longitude:

sin(φ')= cos(φ)*sin(λ) or
sin(Tlatitude)= cos(Latitude)*sin(Longitude).

As a mnemonic for this formula, note that the four items on the right are (in pairs) in alphabetical order: cos, sin and Lat, Lon. It can be written explicitly:

Tlatitude= arcsin(cos(Latitude)*sin(Longitude)).

The route to Tlongitude, λ', is a little more long-winded. It's in the right side triangle with its hypotenuse as the top right edge. We don't have a similar nice value for this hypotenuse which is also awkwardly in the top triangle with Tlatitude as its angle. We can use more easily calculated values by going the other way towards P. Tan(λ') is equal to the front right side divided by the bottom right edge. We note that the front right side is equal to the front left side and that is equal to sin(λ). The bottom triangle has the bottom left edge, cos(φ), as its hypotenuse, so the bottom right edge is cos(φ)*cos(λ).

Putting these together we have tan(λ')= sin(φ)/(cos(φ)*cos(λ)), or:
Tlongitude = arctan{sin(Latitude) divided by [cos(Latitude)*cos(Longitude)]}

with fancy brackets to help to make it clearer. It can also be written as:
Tlongitude = arctan{tan(Latitude) divided by cos(Longitude)}

A bit more of a mouthful, and I can't think of a nice mnemonic.

Although they don't look similar, these results, combined with the Mercator formulae, are easily shown to be equivalent to those given in (3), as official confirmation.

Applying the Mercator formulae and suitable scales to these gives us the (x,y) points needed, so, in my next article I will show how this all provides a basis for understanding the projection the British Ordnance Survey uses for its maps. I am pretty sure that other national surveying bodies use a very similar scheme, although, naturally, I have studied most closely what applies to where I live. There will be an SB program to illustrate it all.

A nice problem, which I haven't yet answered: is there a point P for which both normal latitude equals transverse latitude and normal longitude equals transverse longitude? I think there isn't.

Acknowledgements

I owe a lot of thanks to Geoff Wicks and Jochen Merz for their support and efforts in getting my mathematics on to the page, and to Geoff for supplying the CIA mapping data in a form I could use.

References

(1) to (8) are repeated from my last article; and I have added more comments to some of them.

(1) **http://www.ordnancesurvey.co.uk/**
A wonderful resource for all sorts of map related material.

(2) "Mercator: The man who mapped the planet" by Nicholas Crane ISBN 0-753-81692-X

(3) This is a source for Mercator formulae and

some examples of maps drawn using them, but of course no derivations: Eric W. Weisstein. "Mercator Projection." From MathWorld - A Wolfram Web Resource.
http://mathworld.wolfram.com/MercatorProjection.html (It asks to be cited like that) My TM formulae are easily converted to those in this document, confirming I have got it right.

(4) "Constant Headings and Rhumb Lines", an article that has a beautiful picture of a Loxodrome at:
http://www.mathpages.com/home/kmath502/kmath502.htm

(5) **http://www.uwgb.edu/dutchs/sitemap.htm**
A comprehensive description of UTM.

(6) **ftp://ftp.blm.gov/pub/gis/coast.zip**, 564 Kb.
A List of over 81,000 Co-ordinates I used to draw the World maps. (before Geoff supplied the CIA data)

(7) **http://www.evl.uic.edu/pape/data/WDB_CIA**
World data in a 30Mb .zip file.

(8) **http://home.att.net/~timholt.**
I downloaded the 2m_nav system from this site, about 8.6Mb for Windows -- a comprehensive aid for professional navigators. I am pretty certain he uses (7) for his maps. It has an example of a ship's route from the Adriatic to the Caribbean, nicely illustrating the use of rhumblines to approximate a great circle route across the Atlantic.

(9) Jan Jones "QL SuperBasic The Definitive Handbook" pages 39 and 40.

Machine Logic

by Stephen Poole

In 1931, the Austrian Kurt Gödel proved that mathematics are sometimes imperfect: They are incapable of demonstrating the truth of certain logic suppositions.

Everything began in 1900 when Bertrand Russell gave the example of the Cretan paradox: It is impossible to define the logic of the statement "I am a Liar" in mathematical terms. (If he is a liar, then he is telling the truth, therefore he is not a liar...). Another famous example is the Law of Anarchy: RULE 1, There are no rules. (Therefore there is no Rule 1...). The 'theory of groups' had just been described by Giuseppe Peano. The paradoxes were declared undefinable because they do not contain 'The group of all groups which do not contain themselves'. The whole basis of Scientific certainty collapsed, unless there was some hidden error.

David Hilbert tried to see if there was a universal syntax that was not only coherent but also non -contradictory. He invented meta-language. His ambition was to define a logic which would allow all formula to be derived from simple ones. (Recently this has been developed into 'Lambda Code', which, although long-winded and tedious to elaborate, assists the definition of all computer languages, programmed using just a few symbol types). To see an example of a meta language, just look up the syntax definitions of SuperBasic in the User Guide.

Now we approach the major problem of machine translation: To describe the syntax of a given language we must know beforehand the exact meaning of all words and symbols if we are to avoid ambiguity. Hilbert began by first describing whole number formulae. RULE A: 1+1=2 was true. RULE B: 2+2=5 was false. Gödel insisted on the necessary coherence of the initial logic propositions: So the value of 2 integers(1+1)=2, and the value of 2 integers(1+2)=3 where the value of those 2 integers plus 2 integers, is 2+2=5 (true), but only when the implied logic is clearly defined. Yet Gödel insisted that using the Theory of Groups applied to 'The Liar' paradox, proved that in maths there will always be formula which are indefinable. This was a conceptual bomb for Maths, Linguistics, Psychology and Philosophy, and became known as the Theory of Indecidability. Gödel's system of symbols had the advantage that it could be applied to machines via algorithms, and in 1930 it initially inspired Von Neumann, (one of the major pioneers of modern computing theory), who subsequently abandoned the study of Logic, being disheartened by indecidability! Indeed the indecidability of algorithms meant that it was impossible to predict whether a program would work at all, condemning code-writers to proceed uniquely by trial and error! Desolation in the ranks of programmers!

When I was at Grammar School, our English Language Master taught us grammar as it is generally defined, but with the following important proviso: Grammar cannot be applied without studying underlying logic. For example: "Come here" cannot be understood or translated accurately without knowing its context. At the very least we need to know WHO must come, HOW to come and WHERE 'here' is. For example, "Come here" in French can be "Venez ici" or "Viens ici", as French 'You' can be either the old English 'Ye' or 'Thou'. A machine translation (MT) program may effectively contain this rule, but it is very likely that it will not know from context which is the case. So the program therefore generally chooses one at random! Language is notoriously lacking in precision, as most words are assumed yet missing, only prior knowledge revealing context. An MT system lacks this implied background information and so therefore is incapable of giving unambiguous translations. If we humans don't know the context, we can often deduce it from other elements. Machines don't know where to even start making intelligent guesses outside of grammar. Our brains work a bit like bee hives: Thousands of separate unconscious incoming signals interact with each other until an overall intelligent pattern emerges: apparently-unrelated variables are progressively identified as stable constants, which are then related to more or less fixed verbal and visual symbols. Thus we slowly construct notions from partial elements. Our brains are massively parallel servo-organisms, whereas the machines we have devised as computers cannot assess situations globally, because of their serial step by step operation: They are inherently slugged by a lack of sub-program interactivity. This is why research on neural networks should eventually give pay-offs. But remember, the human brain has some 50,000,000,000 neurones but most machine networks only have several dozens! And human intelligence comes as much from the ever-changing neural cabling as from the contents of the memory centres themselves.

The QL is a great piece of engineering, but a Basic Program Interpreter can only produce results based on the accuracy of the symbolic code you feed into it. (Yes! **BASIC** is by definition "**B**asic **A**ll-purpose **S**YMBOLIC **I**nstruction **C**ode" - (Few people realise that one of its cleverest conceptors was a certain William Gates). - Symbolic SuperBasic's ability to be written using quasi-plain English just reveals how lucky you are that Jan Jones wrote it for the QL). As an illustration of logic applied to machines, I attempted to code 'The Liar' paradox on QPC and came up with the program Gödel_bas, using the rules I learnt at school. The program simulates as accurately as possible how a paradox operates in the brain. On an even simpler level, here is why any proposition may be indecidable:

```
100 loop
110 DEF PROC loop
120 loop
130 END DEFine
```

The program is indecidable because it is infinite as it contains no exit point! (This code will crash the interpreter, as it will rapidly run out of memory...the QL RAM being finite!) On a human level, a problem with no solution may have even more grave consequences, as the victim may go mad pondering on the same never-ending interrogations! I hope this article will help you avoid pitfalls when writing code based on logic structures. (I have found scant few examples in QL literature). The human brain accepts utterly absurd nonsense readily, as it cannot test propositions recursively without making mistakes as it is not infallible, and therefore it ends up coming to semi-random conclusions. Computers may do a similar thing when endless looping on data causes rounding errors to appear and create wild derivation from the predicted bounds until an (erroneous) condition is met.. Mathematicians can and do go astray when they try to reduce formulae to be self-referential, recursive and hence infinite. Much to the pleasure of creationists, Science often disproves itself. But the logic is seldom false, just incomplete in certain contexts.

To avoid the traps of indecidability, propositions must not be incomplete... So, here is how to resolve the 'Anarchy' paradox: RULE 1: 'There are no rules except this one'! And for 'The Liar' paradox: 'I am a Liar, but only when it suits me'! The problem is that mathematicians detest appending conditions to their elegantly simple formulae... Therefore analysis, probability and statistics have become a fundamental part of mathematics, physics and science in general. We live in an inherently curved 4-D universe of Mass-Field theory, described by dozens of pages of eleven-dimensional formulae where solutions are seldom simple.

Computers work thanks to simple binary logic. All things are represented by analogy as 0 or 1. Why by analogy?: Because this is the nature of initial human experience. The complexity of reality demands that we approach solutions by approximation using the old guessing game: Am I getting hotter or colder? (This is how cheetahs and rockets calculate complicated collision courses of intersecting trajectories). In this way we transform discreet binary 0 and 1 into a variable probability range of 0 TO 1. Therefore, so-called 'absolute' machine logic adapts to uncertainty and can represent order, chaos or whatever. Indeed, by attempting to simulate reality as algorithms, we are gradually transforming scientific observations into symbolic information, formulating 'laws' that become more precise as we progress. So with computers we can try to reassemble combina-

RWAP Services NEWS!

**** We have moved ****

See our updated address details below.

We have also acquired more brand new Sinclair QL membranes and another stock of Epson Stylus Colour 850 inkjet printers, so if you need a better printer for your QL, give us a shout.

More news is always available on our website: www.rwapsoftware.co.uk

We are also looking to produce some new hard disk interfaces for the ZX Spectrum and have a few little projects on the drawing board.

Our websites:
http://www.rwapservices.co.uk (General site)
http://www.rwapsoftware.co.uk (Sinclair computer second hand and new items)
http://www.rwapadventures.com (Adventure Programs)
http://www.internetbusinessangels.com (Guidance on setting up online businesses).

New Products!

QWORD 1.0 NOW WITH DIGITAL SOUND ON QPC2!

The wait is now over! Q-Word version 1 is finally available!

Platforms:
QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word	£20.00
All versions with P-Word	£30.00

Notes:
Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.
The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

TALENT COMPUTER SYSTEMS for **Windows**

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows	ea. £ 10.00

(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

Games Currently Available from www.rwapadventures.com

The Lost Kingdom of Zkul
West
The Prawn
Return to Eden

Replacement Sinclair QL Keyboard Membranes

We always have a stock of brand new Keyboard Membranes (and keyboard parts) for the original Sinclair QL, so if you have some keys which no longer work, just give us a call.

Cost is only £18.50 plus £2.75 post and packing.

Second Hand Items - Huge Range Available

We stock a wide range of books, hardware and software for the Sinclair QL, Z88 and ZX Spectrum, including disk interfaces, memory expansion and microdrive cartridges. If there is anything you need - have a look at www.rwapsoftware.co.uk (or ring us with details of your requirements).

We are always happy to help.

Old Favourites!

Utilities

SBASIC / SuperBASIC Reference Manual on CD	**£ 20.00**
Sidewriter v1.08 *Landscape Printing (EPSON printers)*	£ 10.00
ImageD v1.03 *3D object generator*	£ 10.00
Q-Help v1.06 *Superbasic On-Screen help system*	£ 10.00
Q-Index v1.05 *Keyword-to-topic finder*	£ 5.00
ProForma ESC/P2 Drivers v1.04 for ProWeSs *Printer Driver*	£ 8.00

Applications

Flashback SE v2.03 (upgrade only) *Database*	£ 5.00
QL Cash Trader v3.7 *Accounting/Finance*	£ 5.00
QL Payroll v3.5 *Accounting/Finance*	£ 5.00
QL Genealogist v3.26 *Genealogy*	£ 20.00
Genealogy for Windows	£ 50.00
QL Genealogist to Windows version upgrade	£ 25.00
QL Cosmos v2.04 *Planetarium*	£ 5.00
Q-Route v2.00 *Route Finding*	£ 25.00
Upgrade from v1.xx	£ 5.00
Britain map v1.11	£ 2.00
BIG Britain map (needs 2Mb) v2.03	£ 5.00
Various Britain Area maps (ask for details)	ea. £ 2.00
Ireland map v1.00	£ 5.00
Belgium map v1.01	£ 2.00
Catalonia map v1.03	£ 2.00
P-Word UK English Dictionary (500.000 words!) *Dictionary*	£ 15.00

Leisure

Return to Eden v3.08 *Adventure*	£ 10.00
Nemesis MkII v2.03 *Adventure*	£ 8.00
The Prawn v2.01 *Adventure*	£ 8.00
Horrorday v3.1 *Adventure*	£ 8.00
West v2.00 *Adventure*	£ 5.00
The Lost Kingdom of Zkul v2.01 *Adventure*	£ 5.00
All 6 games above	£ 25.00
D-Day MkII v3.04 *Strategy/War Simulation*	£ 10.00
Grey Wolf v1.08 *Graphical Submarine Simulation*	£ 8.00
War in the East MkII v1.24 (upgrade only) *Strategy/War Simulation*	£ 5.00
Open Golf v5.20 *Sports Simulation*	£ 8.00
QuizMaster II v2.07 *Quiz*	£ 5.00
Stone Raider II v2.00 *Arcade Game*	£ 5.00
Hoverzone v1.2 *Arcade Game*	£ 5.00
Deathstrike v1.5 *Arcade Game*	£ 5.00
Flightdeck v1.0 *Flight Simulation*	£ 10.00
All 6 games above (Open Golf, QuizMaster II,Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00

Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

RWAP Services

3 Dale View Court, Fulford, Stoke-On-Trent, Staffordshire ST11 9BA TEL: (+44) 1782 398143 From the UK Dial: 01782 398143
Website: http://www.rwapsoftware.co.uk
Email: sales@rwapsoftware.co.uk

We Accept Payment using:

PayPal VERIFIED NOCHEX EMAIL MONEY PREFERRED PAYMENT (Cheques in £ sterling made payable to R. Mellor) tions of those symbols to make interpolations which predict other properties of the universe. The wheel of logic has turned full circle. There is a program for E=mc2 but it was conceived strictly without the help of any machine logic. Einstein was one of the last great thinkers who came to very highly accurate conclusions using the simple tools of pencil and paper to aid insight.

It has been seen that Machine logic is useful as an aid to making decisions. Climate Change studies demand that a very great many specialists cooperate on making observations and forecasts and this is one of the domains that massively need the assistance of computers because of the sheer bulk of calculations involved. Climate research in relation to Global Warming is responsible for some of the most important knowledge that will affect the future not only of mankind, but also the future of all biodiversity on the planet. So computer research has now reached the point where we are becoming increasingly dependent on it. But as always we are faced with the challenge that to be accurate, logic must be complete. So there is plenty of scope for scientific study by future generations, that is, if we act quickly enough to ensure that we leave an inhabitable planet for our descendants to live on. Without man, the whole logical question of whether natural law or intelligent design exist is pretty futile from an anthropocentric point of view.

```
100 ::
110 REMark Gödel_bas, by S.Poole, v27nov07
120  REMark 'I am a Liar' said the Liar...
130   DIM Truth$(2,6): Truth$(1)='True?': Truth$(2)='False?'
140 :
150 REMark Strictly speaking, using NOT, True=1 or False=0.
160  REMark But who ever knew a liar who never tells the truth?
170   truth=-1: REMark So begin with twisted logic (as -1 is not 1).
180 :
190 REMark Absurd loop:
200 REPeat arf_arf
210    truth=NOT(truth): REMark This is a lie.
220    REMark Now see if original proposition is true or stay confused:
230    IF truth=-1: EXIT arf_arf: ELSE PRINT Truth$(RND(1 TO 2))!!: PAUSE 5
240 END REPeat arf_arf: PRINT 'Eureka!'
250 :
260  REMark The proposition is indecidable...
270   REMark Because incomplete and therefor infinite!
280    REMark Iteration & Recursion needs a valid exit point.
290     REMark So don't underdefine self-referential expressions.
300 ::
```

Byts of Wood

by Roy Wood

To start off I would like to apologise for not being able to attend the recent Quanta AGM in Manchester. A combination of factors conspired to keep me away. The first was related to my rock band which had a late show on the Friday before but the main clincher was a course, which I had booked and paid for months ago, was moved forward two weeks and landed slap bang on the main day of the show. Since I have now taken over as Diving Officer for the Sussex BSAC I had to attend.

Even before the course change, however, I was thinking long and hard about the viability of travelling to Manchester. Fuel costs these days are high and I was likely to be the only person travelling up from Brighton. In the past I have shared some of the costs with Tony Firshman but I knew he was unable to go. Past shows have, on many occasions made enough money for me to be able to write off my expenses against sales but the last couple of years have seen a steady decline in income - making these trips more and more uneconomical. For Tony and I this is a real problem.

Of course this is not the fault of the people who come to the shows or the fault of the organisers. We all do these things in good faith and try to keep it all running but there does come a time when we have to reflect upon the reasoning behind it all.

Not Waving Goodbye

Now it is the very nature of all things in physics and nature to behave in waves. The first days of the QL, when it was announced to the public, were a high crest followed by a bit of a trough when the would be owners of this new machine found it was not available - or even ready. Then release day, another high point and down again when the new owners found the dongle. The cycles continue. There are many high points such as the introduction of the Gold Card, the Super Gold Card, the Qubide, the Aurora board and the Q40. There are also low points such as the end of QL World, Qubbesoft, Miracle Systems and the many software writers who left the scene to pursue other goals or write for other systems. Such is life, nothing is static.

It is also in the nature of things that all order descends to entropy or towards a state of growing disorganisation as it loses energy. This has also happened in the QLs history. This is sort of equivalent to the coming of Spring when all the buds start to poke their way out and you feel all energised about wanting to go out into the garden and mow the lawn, dig out the weeds and get everything sorted. After a while you tend to go out and lie in a hammock and look at the flowers.

Each time the QL scene has dropped back in to this state there has been a bit of a push to get it started again and it sets off in a new direction. Lots of events have kick started us towards this in the past, the arrival of the Q40 was one such recent(ish) event and I think, in a modest way, the start of QBranch 13 years ago. New hardware and new traders bring fresh energy to bear on things. I was certainly optimistic when I started out in 1995 and I was similarly motivated when I took on selling the Q40 too.

Fade to....Grey?

One of the events that caused less of a stir than they should have done was the arrival of the colour drivers. Admittedly these took a while to arrive in a usable form and it was apparent that, while these colours were amazing for people who had never used a computer other than the QL, they were a bit old hat for anyone who had a PC. I suppose it was inevitable that what came out of the advent of the new colour facilities was so similar to the desktop as seen on a PC or a MAC. In hindsight I suppose we should really have tried to forge a new direction for ourselves but it is hard to see what that direction could be. Even Microsoft with all its design teams and resources only manages to round a corner or two and change the shading a bit when it brings out a new system.

The big downside of the colour drivers has always been the amount of resources they need to get the most out of them. The standard QL-itself cannot handle them at all. An Aurora board can make a pretty stab at the lower reaches of the colour capabilities but at the expense of so much memory usage that there is little power left to run the computer. Only QPC and the Q40/Q60 are able to adequately do the colour drivers enough justice for all the work they took. A shame really because there was so much promise and capacity there. Marcel's sterling work adapting Easyptr to be able to program the new colour schemes made it so easy to make nice looking programs but somehow few people take advantage of it.

We Need More Power Captain

Of course the big stumbling block in all of this is the lack of raw power in any of the QL hardware solutions. We have discussed many times in this column how hard it is to make new hardware and how much money and effort have to go into it. Any hardware solution to this problem would need to be tailored to the original black box itself. The Aurora was a brilliant piece of kit but, sadly, sold too few units to become ubiquitous. If you build an add on for the original QL you wind up without the the higher power graphics available from the Aurora. Trouble is, if you base it around the Aurora then you won't have enough sales to warrant the work so where do you start?

And then there is the hardware versus emulation argument. Is a Q60 a QL? well no it is a different board using a QL-based operating system so it is not really any different to QPC in that it is a QL-system sitting on top of a different piece of hardware. Admittedly the Q60 is specifically designed for this and QPC is an emulation on an operating system so neither is a QL-in the real sense of the concept but QPC is more efficient because it harnesses the power of the modern PC beneath it.

Software

There has been precious little new software in the last few years and that lack of innovation has had an effect of the QL Scene too. When I was asked about QL-workshops recently I had too reply that I saw little point in them because there was nothing new to promote or sell. For a trader the journey costs have to be covered by the sales and that has not been the case in the last couple of years. This magazine always tried not to present a negative outlook or express negative views but I have always felt that was wrong. We should not have our heads in the sand and deny all negative aspects of the way the QL scene has deteriorated we should be looking at negative comments as a way of improving the areas we do worst in.

There have been recent triumphs. QPCPrint was one of these, enabling users of the QPC system to print to any modern printer but it only works for QPC and on a PC. If we adopt the 'heads in the sand' position we may find that, in the end, all hardware, printers/drives/keyboards etc. are no longer capable of connection to a QL. There may be no commercial hardware available to take it's place and we will be left with emulators alone.

There has been recent talk on the QL email user group of a new Qubide being made. I was quite intrigued by this, since it implied that there were enough users of native hardware to take it up. I do not know too much about this but, one of my customers, who was in touch with the man who is doing the work, wanted to buy a set of Gals and ROMs from me to use on it. If they are using the same GALs and ROM as the original then the new one will not be any more capable than the previous one of handling modern drives. He was, though, asking if anyone had any ideas for the new Qubide so maybe he is also planning improvements to the original design.

It was always Nasta's intention to make a new Qubide first and, I must confess I always saw that as being the wrong way round. Surely the important need for anyone using a native hardware system is more speed and more RAM to make the colour drivers and some of the more intensely CPU hungry programs work. None of these come even close to the demands of a PC program however.

Menus

Jochen, just to prove me wrong, had recently done some work on the Menu extensions. I have inserted his changes list and comments below:

Version 7.67

LIST Autosize correct if selection keys given ;
LIST Long titles fit even in small menus ;
RPER Timeout for RPTR loop in upper word of colourway parameter ;
ITSL Timeout added as for RPER; return value if timed out is -3 ;
RSTR Bit 6 set: negative sign allowed for numbers. ;
ITSL Window move implemented.

To give you an idea about the timeout:

```
1  timeout=10:PRINT  ITEM_SELECT('Please Select','You have '&timeout&' seconds to choose!',A,B,C, , ,(timeout*50)*65536)
```

I will try out the Timeout feature in my new version of the invoice program but it would be nice for menus to know when the pointer leaves the window and puts itself to sleep, or back to its state before it was clicked on, too. It is hard not to judge the programs we use on the QL against those we use in every day work life on a PC or MAC. Little inconveniences like having to ESCape from a menu if you call it accidentally are annoying after a while. Here's a little idea for the Timeout function. How about popping up a small window with a programmable message like 'Closing in X seconds' and having it count down. That would be neat.

Jim Hunkins has also said he has restarted work on QDT files menus so this could lead to some more new developments and, by far the most active QL Trader at the moment has to be Rich Mellor who has been trying to push Quanta and QL Today subscriptions and sending out mountains of email offers.

Let's hope this is the tip of an larger iceberg - or even a new wave.

Don't let my previous comments lead you to believe that I am against having workshops or packing my bags and heading for the hills. I would, however, like to see some progress, some movement and something new to get the interest flowing again.

AND NOW, number whatever in a series of computer related definitions.....

Dip Switch - small flagellation device used to get bathers to go into cold water

Jumper - someone committing suicide after failing to set up an old motherboard

CPU fan - someone who is really into CPUs

Rear Fan - One of Jay-Lo's admirers

Digital Video - having a finger over the lens when filming

Annalogue video - Karenina's Video Diary

Patch Bay - Sewing basket

Server - Paul Burrel's old job

Creche - what computers do in Hampstead

Optical Disc - contact lens

Powerpoint Presentation - showing people where to plug the computer in

Microsoft - swear word in Ubuntu

Nintendo Wii - burning sensation in the bladder which tells you you have spent too long playing games

Web-Bot - Medical condition caused by sitting down too long surfing the Internet

Screen Saver - metal grill just in front of the monitor to prevent the user putting his fist through it at times of stress.

Firewall - what happens when the computer power supply blows up and sets fire to that stack of invoices and copies of QL Today you have been meaning to put away

Remote Control - someone in the next room telling you to turn off that computer and come to dinner

Denial of Service Attack - when your wife complains that you don't come to bed till after 2 oclock because you are sitting at the computer.

CAT5 - 5th feline owned by someone to lazy to give them names

Data Collision - When two girlfriends turn up at the same time

EPROM - Programme of light classical music for people from Yorkshire

E-EPROM - Same as above but as a podcast

Hot-Swap - Getting lucky at a swingers party

Wi-Fi - Elizabethan exclamation

Graphics Tablet - drug to keep you calm when trying to use Photoshop

Hand-Held PC - Special Constable

Motherboard - The thing she does the ironing on.

Thin Clients - customers of a computer dealer who sells IT to supermodels

Quantity Surveyor - Workshop beverage inspector

Nanosecond - your stepfather s mother.

Menus

by Jochen Merz

As Roy mentioned, I was busy at the last Eindhoven show. As promised to Bob Spelten, I was looking at some problems he had with Menus, and I also looked at some improvements wished by various users.

The queues at my desk known from the past did not exist anymore, unfortunately (but Roy explained this in his current BOW as well), so I started looking at the sources. Haven't looked at them for many years, but surprisingly, I got into programming mode quite fast - and produced small improvements very quickly, with a bit of debugging. Bob tested the new versions immediately. Francois van Emelen watched all this very interested, and I thought he (sort of) understood the assembler programming, but he told me later, he didn't. Well, thanks for the support and the interest in general.

There is one little thing I would like to fix before I release this version ... and I hope to be able to do it at the next Eindhoven show, next month.

I will notify about a new version on my website SMSQ.J-M-S.COM and also put it into the J-M-S download area for registered users. Please do not ask in the meantime.

I have to admit: programming again and seeing immediate results was very enjoyable - I really noticed that day how much I missed it. Pity, I rarely find the time. There are so many things I'd like to do. Maybe at the forthcoming QL meetings ... see you there!

And now, after you have read through this issue of QL Today and hopefully enjoyed reading it - please renew now if you have not done it yet. Renewal form is attached, but you can also send an email, fax it etc. - we are flexible, as you know!

The QL Show Agenda

QL Meetings in Eindhoven (NL)

Saturday, 14th of June 2008, 11:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

Same venue as always. (J-M-S plans to be there, as always - but please visit **SMSQ.J-M-S.COM** to see if he can make it - health issues might be getting in the way). There is another fixed date for the autumn show, so that you can add to your agenda.

Saturday, 18th of October 2008, 11:00 to 16:00

QL Meeting in Italy

Sunday, 15th of June 2008 at DTM, Modena, Italy

End of April, Davide Santachiara announced this meeting on QL-Users newsgroup. Although it is very short notice, we try to support this by informing here - hoping we get the magazine out to our readers earlier than scheduled.

On Sunday June 15th, there will be a very informal get together with Italian QL users in Modena (40 Km far from Bologna where the nearest airport is located). This time the meeting will be held in a couple of meeting rooms gently made available by the company for which I work. At launch time we will move to a trattoria very close to the meeting place to have some typical dishes of Emilia.

Instructions to reach the meeting place can be easily retrieved from the company web site:
http://www.dtm.it/home.php?id=40201

Modena is a nice city located in Emilia Romagna, Northern Italy:
http://turismo.comune.modena.it/index-en.php

the Cathedral and the Piazza Grande are Unesco Heritage sites
http://www.sitiunesco.it/index.phtml?id=577

Close to Modena are located Parma and Bologna (45 Km), Florence and Venice can also be easily reached by car or train (1:30 - 2:30). Closest airport are Parma (45 Km), Bologna (45 Km), Verona (100 Km), Forlì (120 Km). Ryan Air flies to Parma and Forlì.

The meeting will start at 10:00 end it will end at 17:00, lunch time from 13:00 to 14:30. If you would like to attend, for help on accommodation etcetera please get in contact with Davide Santachiara: **d.santachiara@libero.it** *- cell phone* **+39 328 9454484.**

The Next Issue & Volume

We plan to have the next issue (Issue 1 of Volume 13) ready for you at the middle of September. As always, it depends on how quickly we get reviews, articles etc.

We look forward to our next volume, and thank you very much for renewing and staying with us, if you haven't already done it yet.

Great to see that we, the QL community, are still so active and alive, even without many shows.